

DESARROLLO Y VALIDACIÓN DE UN MODELO DINÁMICO PARA UNA PILA DE COMBUSTIBLE TIPO PEM

Alejandro J. del Real, Alicia Arce y Carlos Bordons

Departamento de Ingeniería de Sistemas y Automática

Escuela Superior de Ingenieros. Camino de los Descubrimientos s/n. 41092 Sevilla

bordons@esi.us.es

Resumen

El objetivo de este trabajo es realizar un modelo dinámico detallado de una pila de combustible tipo PEM de 1.2 kW de potencia nominal. El modelo desarrollado incluye efectos como el 'flooding' y la dinámica de la temperatura y es de utilidad para poder diseñar y ensayar controles tanto de la válvula de purga como de la refrigeración de la pila mediante un ventilador. Se ha desarrollado un novedoso tratamiento de la ecuación experimental que modela la curva de polarización que simplifica considerablemente su caracterización. Por último el modelo realizado ha sido validado con datos tomados de una pila real.

Palabras clave: Modelado y simulación, Pilas de combustible.

1. INTRODUCCIÓN

Las pilas de combustible tipo PEM son dispositivos electroquímicos que transforman la energía química de un combustible (hidrógeno) directamente en energía eléctrica mediante su reacción con el oxígeno del aire, teniendo como residuos agua y calor. Por esta razón y por las bajas temperaturas de funcionamiento y su rápida puesta en marcha, las pilas PEM se consideran buenas candidatas para en el futuro sustituir a las fuentes convencionales de energías actuales tanto en aplicaciones residenciales como en automovilísticas.

Para poder desarrollar buenas técnicas de control, factor decisivo en la obtención de un funcionamiento óptimo de la pila, es necesario tener un buen modelo donde poder simularlas y además a partir del cual se pueden diseñar estrategias más avanzadas, por ello hemos desarrollado un modelo avanzado de pilas de combustible que tenga en cuenta todos los fenómenos que se desean controlar en la pila y además represente todas las etapas de funcionamiento de la pila tanto los arranques como las paradas, importantes en aplicaciones automovilísticas en las cuales se deben desarrollar técnicas específicas que regulen estas etapas con seguridad.

Se ha realizado un estudio del estado del arte de los modelos existentes en la bibliografía de las pilas de combustibles y no se ha encontrado un modelo que aunase en uno solo todos los fenómenos que deben ser controlados como el encharcamiento *flooding* o la evolución de la temperatura de la pila aparte de los comunes como el suministro de hidrógeno y oxígeno procedentes del aire. Los modelos que se encuentran se centran específicamente en uno de estos fenómenos resultando complejos modelos que están más orientados al diseño de células que al control de éstas. El modelo que presentamos está dividido en tres partes claramente diferenciadas: la dinámica de los fluidos, la dinámica térmica y las características eléctricas. Se caracteriza principalmente porque modela todos los fenómenos de la pila con la suficiente sencillez necesaria en las aplicaciones de control.

En las primeras secciones se presenta cada una de las partes en las que se ha dividido el modelo, destacando las mejoras e innovaciones desarrolladas para cada una de ellas. En la siguiente sección se presenta los modelos de los equipos auxiliares que deben ser incorporados para que la pila de combustible pueda funcionar y por último se presenta la validación del modelo desarrollado con los datos obtenidos de una pila de combustible PEM de 1.2kW.

2. MODELO FLUIDODINÁMICO

En esta parte se modelan las dinámicas de los fluidos implicados en el proceso en su paso por la célula de combustible. Las células están compuestas por dos canales, ánodo y cátodo, separados por una membrana (MEA Membrane Electrode Assembly). Además incluyen dos capas de difusión construidas de material poroso e hidrófobo, situadas entre la MEA y los canales, facilitando la difusión y contacto de los gases con la membrana. Estas células se conectan en serie para formar las pilas o 'Stacks' y así alcanzar la potencia requerida.

La membrana, siendo impermeable, permite la difusión del agua en forma de vapor, creándose un flujo neto de transferencia entre el cátodo y el ánodo.

do por dos efectos: por un lado, los protones disociados del hidrógeno en el ánodo, en su migración hacia el cátodo, arrastran moléculas de agua. Este fenómeno es conocido como *electro-osmotic drag*. Por otro lado, debido a la diferencia de concentraciones, el vapor de agua se desplaza desde regiones de menor concentración a otras mayores. Este efecto es llamado *back diffusion*. Cuando la producción o transporte de agua supera la posibilidad del vapor de difundirse desde las capas de difusión hasta los canales de caudal, el vapor satura y condensa líquido. El agua entonces se va acumulando en las capas de difusión hasta que la presión capilar hace que fluya hacia los canales de caudal. Este agua ocupa parte del volumen de poro de las capas de difusión primero y luego, al llegar a la superficie de éstas, reduce el área efectiva de intercambio de gases, dificultándose de esta manera su difusión. Este fenómeno, que hace caer el rendimiento de la célula de combustible, es conocido como encharcamiento o *flooding*. El tratamiento de este fenómeno es una novedad en nuestro modelo.

Los fluidos implicados en este modelo son el hidrógeno, oxígeno, nitrógeno y agua en forma líquida y vapor. En [1] se presenta una solución del flujo tridimensional basada en las ecuaciones de Navier-Stokes, usando el programa de cálculo FLUENT, pero desde el punto de vista computacional para aplicaciones de control esta solución es inviable. En [2] se presenta un estudio centrado en los fenómenos que tienen lugar en las capas de difusión, teniendo en cuenta la variable espacial perpendicular a dichas capas. En [3] se simplifica esto último tomando un modelo de la capa de difusión de parámetros concentrados.

En este trabajo se ha optado por el último enfoque, pero incluyendo algunas variaciones. Primero, para reducir al máximo el costo de cálculo, se ha tomado un único volumen de control para cada capa de difusión, en lugar de tres. Esto imposibilita el conocimiento de las variables en el interior de dichas capas, pero permite evaluar su valor medio. En los canales de cátodo y ánodo se incluyen también dinámicas de evaporación del agua. Además, el tratamiento del agua líquida se ha modificado suponiendo que además de salir de la pila de combustible en forma de vapor, una cierta cantidad sale en forma líquida por arrastre mecánico con los caudales de salida. Esto permite un estudio más fiel de los fenómenos que ocurren durante una purga del ánodo puesto que se puede suponer entonces que el agua líquida acumulada en la superficie de la capa de difusión del ánodo es así como se elimina.

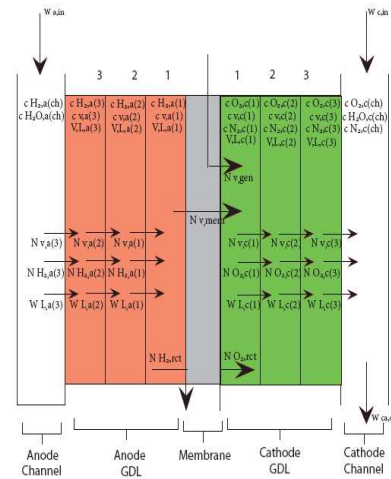


Figura 1: Discretización en volúmenes de control de una célula de combustible

Entrando un poco más en detalle de esta parte del modelo, se puede dividir en cinco bloques interconectados, correspondientes cada uno a los volúmenes de control de canales y capas difusoras de cátodo y ánodo y al transporte de especies a través de la membrana. Es importante tener en cuenta los criterios de signo que se muestran en la siguiente figura.

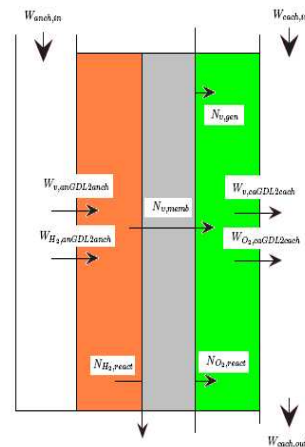


Figura 2: Criterio de signos

2.1. Canal del cátodo

El caudal de entrada al canal es aire proveniente del sistema auxiliar de acondicionamiento que tenga acoplado la pila de combustible. El caudal linda a su vez con la capa de difusión del cátodo y entre ellos habrá un intercambio de agua y de oxígeno. El flujo de agua depende de la diferencia de concentraciones entre ambos volúmenes de control y es calculado en las capas difusoras. Se han despreciado las posibles diferencias de presión de oxígeno entre el canal y la capa difuso-

ra del cátodo, ya que estudios como [2] muestra son muy pequeñas y las dinámicas de intercambio por diferencia de concentraciones entre ambos volúmenes de control muy rápidas. Se utilizó [7] para describir las dinámicas de evaporación y condensación en el canal. Se ha tenido en cuenta también la restricción lógica de no evaporación de más cantidad de agua que la líquida disponible. Las ecuaciones dinámicas que aporta este bloque son las siguientes:

$$\frac{dm_{l,cach}}{dt} = 0 \quad (1)$$

$$\frac{dm_{v,cach}}{dt} = W_{v,cach,in} - W_{v,cach,out} + W_{v,caGDL2cach} + W_{evap,cach} \quad (2)$$

$$\frac{dm_{O_2,cach}}{dt} = W_{O_2,cach,in} - W_{O_2,cach,out} - W_{O_2,caGDL2cach} \quad (3)$$

$$\frac{dm_{N_2,cach}}{dt} = W_{N_2,cach,in} - W_{N_2,cach,out} \quad (4)$$

$$m_{ma,cach} = m_{O_2,cach} + m_{N_2,cach} + m_{v,ca} \quad (5)$$

2.2. Canal del ánodo

El bloque correspondiente al canal del ánodo es análogo al del cátodo. En este caso el caudal de alimentación será hidrógeno seco.

Durante una purga, los caudales de salida del ánodo son calculados como sigue, siendo t_{purga} un tiempo obtenido experimentalmente en el que la mayor parte del agua líquida condensada en la superficie del canal de difusión del ánodo es arrastrada por el caudal del gas de purga. Cuando la válvula de purga esté cerrada, $K_{anch,out} = 0$, tomando esta variable un valor mayor que cero cuando se encuentra abierta. Destacamos la ecuación relacionada con la purga.

$$W_{l,anch,out} = \frac{m_{l,anch}}{t_{purga}} \quad \text{si } W_{ma,anch,out} > 0 \quad (6)$$

2.3. Capas difusoras de cátodo y ánodo

Las capas difusoras de cátodo y ánodo se modelan cada una como un único volumen de control, siguiendo el criterio de signos anteriormente explicado para los caudales. Las ecuaciones que modelan los fenómenos que en ellas tienen lugar pueden dividirse en dos bloques, correspondientes uno a la fase gaseosa y otro a la fase líquida. El problema queda simplificado al estudio de los gradientes de concentración del agua en forma de vapor y la presión capilar en forma líquida, además de las dinámicas de condensación.

Las ecuaciones dinámicas que se añaden en la

parte de fase gaseosa son las siguientes:

$$\frac{dp_{v,caGDL}}{dt} = RT_{st} \frac{N_{v,gen} + N_{v,memb} - N_{v,ca}}{t_{GDL}} + RT_{st} R_{evap,ca} \quad (7)$$

$$\frac{dp_{v,anGDL}}{dt} = RT_{st} \frac{N_{v,an} - N_{v,memb}}{t_{GDL}} + RT_{st} R_{evap,an} \quad (8)$$

Por otro lado la fase líquida aporta las siguientes dinámicas:

$$\rho_l \frac{dV_{l,ca}}{dt} = -W_{l,ca} - R_{evap,ca} M_v \varepsilon V_{GDL} \quad (9)$$

$$\rho_l \frac{dV_{l,an}}{dt} = W_{l,an} - R_{evap,an} M_v \varepsilon V_{GDL} \quad (10)$$

2.4. Membrana

La membrana, al ser impermeable, no deja pasar agua líquida pero sí admite la difusión de gases a través de ella. Así, permite la combinación de hidrógeno y oxígeno, pero a su vez deja paso al vapor de agua y al nitrógeno. La circulación de nitrógeno se produce desde el cátodo hacia el ánodo por diferencias de concentración al estar alimentado este último por hidrógeno puro. Este fenómeno se despreciará por ser muy lento y entonces afectar levemente a la presión total del ánodo, y también por ser un gas inerte que no afecta en la reacción. Además, este nitrógeno presente en el ánodo nunca llega a ser una cantidad importante puesto que se elimina durante las purgas. La circulación del vapor de agua es de mayor complejidad, y se debe a los dos fenómenos descritos en la introducción de *electro-osmotic drag* y *back diffusion*. Así, el caudal molar circulante a través de la membrana será:

$$N_{v,memb} = n_d \frac{i}{F} - \alpha_w D_w \frac{c_{v,ca} - c_{v,an}}{t_{memb}} \quad (11)$$

donde $i = I_{st}/A_{fc}$ es la densidad de corriente, n_d es el coeficiente de arrastre electro-osmótico, D_w es el coeficiente de difusión del vapor en la membrana, t_{memb} es el espesor de la membrana y $c_{v,ca}$ y $c_{v,an}$ son las concentraciones de agua a ambos lados de la membrana. El coeficiente α_w es un parámetro de corrección experimental, puesto que todas las constantes relacionadas con la membrana provienen de bibliografía donde se obtuvieron a partir de membranas más antiguas que las actuales.

3. MODELADO TÉRMICO

La parte del modelo que considera este efecto se basa principalmente en [4], aunque hay en la bibliografía más artículos que tratan el problema de manera similar [5]. En [4], se presenta primero un

estudio detallado del calor generado por la reacción electroquímica a causa de las irreversibilidades del sistema y su transmisión al ambiente y al caudal de fluido refrigerante, para después simplificar la formulación matemática obviando los efectos menos importantes. La pila de combustible allí modelada suponía humidificación externa y agua líquida como refrigerante, siendo la utilizada para el modelo aquí presentado autohumidificada y con aire atmosférico como fluido refrigerante. Lo que se ha hecho es partir del modelo ya simplificado, y adaptarlo para las condiciones particulares de la pila aquí estudiada.

Para obtener el modelo térmico, se hace un balance energético en el que se tienen en cuenta la energía producida por la reacción química de formación de agua (que se supone en forma de vapor) \dot{H}_{reac} , la energía aprovechada en forma de electricidad P_{elec} y la cantidad de calor evacuado por radiación $\dot{Q}_{rad,B2Amb}$ y por convección, tanto natural como forzada $\dot{Q}_{conv,B2Amb}$. Para este modelo, la disipación activa de calor se realiza por convección forzada mediante un ventilador. El balance energético queda entonces como:

$$m_{st}C_{st}\frac{dT_{st}}{dt} = \dot{H}_{reac} - P_{elect} - \dot{Q}_{rad,B2Amb} - \dot{Q}_{conv,B2Amb} \quad (12)$$

La energía aprovechada en forma de electricidad será:

$$P_{elec} = V_{st}I_{st} \quad (13)$$

4. MODELADO ELÉCTRICO

las dinámicas eléctricas son despreciables frente a las demás. Es por esto los estudios presentes en la bibliografía presentan las variaciones estáticas del voltaje de las células frente las densidades de corriente suministradas. En [8] se presenta una ecuación con parámetros experimentales para modelar la forma de estas curvas de polarización, que recoge bien el funcionamiento estacionario pero no el transitorio. En [6] se desarrolla una ecuación con más parámetros, buscados a partir de datos experimentales, que modela mejor los cambios en la curva de polarización ante variaciones de la temperatura y presiones parciales de los fluidos reaccionantes (hidrógeno y, sobre todo, oxígeno). La búsqueda de parámetros a partir de esos datos experimentales ha de realizarse allí mediante algoritmos de optimización. Quizás la aportación más importante de este trabajo sea el nuevo tratamiento matemático propuesto para hallar la curva de

polarización. Siguiendo la misma filosofía que en [6], se ha simplificado al máximo esta expresión matemática, suponiendo ahora variaciones lineales con la temperatura y logarítmicas con las presiones de oxígeno e hidrógeno, desacoplando estos dos fenómenos y escalando la curva en función de ellos en lugar de escalarla y además modificar su forma. Esta simplificación, además de modelar fielmente el comportamiento estático eléctrico real, hace que sea posible, mediante consideraciones geométricas, encontrar un algoritmo algebraico directo que calcule los parámetros a partir de unos pocos datos experimentales. Las implicaciones de esto son muy grandes, puesto que se abre la puerta al cálculo en línea de curvas de polarización muy exactas, que representan el voltaje de la célula frente a la densidad de corriente, $i = I_{st}/A_{fc}$, llamadas *curvas de polarización*, y que dependen de varios factores:

Intensidad El voltaje a circuito abierto (v_0) decae a medida que la célula de combustible va proporcionando un mayor valor de intensidad. Así, en una primera etapa y hasta una determinada intensidad, dominan las llamadas pérdidas por activación (v_{act}), que provienen de la necesidad de mover los electrones y romper y formar los enlaces químicos en ánodo y cátodo. En una segunda etapa, cuando la intensidad va siendo mayor, las pérdidas dominantes son las óhmicas (v_{ohm}), debidas a la resistencia de la membrana a la transferencia de protones y a las resistencias eléctricas internas de los electrodos a la transferencia de electrones. A elevada intensidad, a niveles de potencia máxima de las células, las pérdidas por concentración (v_{conc}) hacen caer rápidamente el voltaje, debido a las irreversibilidades que se producen a altos niveles de consumo de los reactivos. Estas últimas caídas, aunque contempladas en la ecuación propuesta de la curva, serán muy leves en el rango de funcionamiento normal de las pilas de combustible, pues operar a esos rangos de intensidad degrada rápidamente las membranas. En (Fig. 3) puede verse un ejemplo de curva de polarización correspondiente a la pila de combustible de 1.2 kW particularizada en este trabajo, con la contribución desglosada de las pérdidas antes comentadas.

Presiones parciales de los reactivos La presión parcial de oxígeno en el cátodo y de hidrógeno en el ánodo influyen de manera notable en el voltaje que dan las células de combustible. A medida que aumenta la presión de alguno de estos gases, el voltaje se incrementa para cualquier valor de intensidad. Dado que en las configuraciones típicas

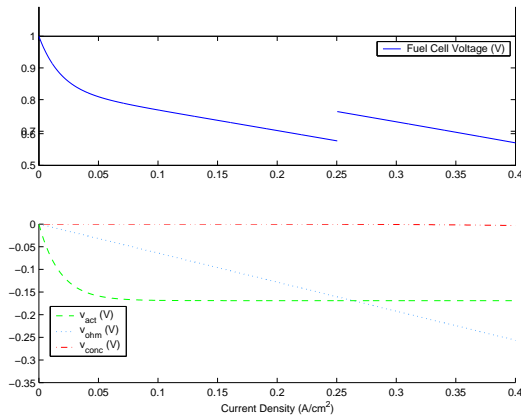


Figura 3: Curva de polarización desglosada

en *dead-end* la presión parcial de hidrógeno permanece aproximadamente constante, es la presión parcial de oxígeno en el cátodo la que merece mayor atención, puesto que es la que sufre mayores fluctuaciones ante cambios en la demanda de energía de las pilas de combustible, al estar en dependencia con los sistemas auxiliares de suministro de aire al cátodo.

Temperatura La temperatura a la que las células de combustible operan tiene una doble influencia en el voltaje. Por una parte, cambios en la temperatura influyen en las presiones de los gases en el interior de cátodo y ánodo. Por otra, las propiedades de la membrana cambian de manera que un aumento de temperatura conduce a un aumento de voltaje para cualquier intensidad. Es por esta razón importante a la hora de realizar una caracterización experimental de las células de combustible distinguir entre estos dos efectos. Dada la dificultad técnica de incorporar sensores de presión en el interior de los canales de caudal de cátodo y ánodo, es entonces cuando es importante el modelo matemático para obtener valores simulados de estas presiones, y así poder discriminar la variación del voltaje con la temperatura de su variación con las presiones parciales de los reactivos.

Cantidad de agua condensada Como se ha comentado anteriormente, la presencia de agua líquida ocupando los poros de las capas difusoras y su superficie provocan caídas en el voltaje y por tanto en el rendimiento de las células de combustible. En el cátodo este fenómeno no es importante al ser este agua arrastrada con el caudal de aire de alimentación, pero es en el ánodo donde se acumula hasta que se elimina por medio de una

purga. Siguiendo el mismo desarrollo que en [3], la solución para modelar este efecto consiste en suponer que a medida que aumenta la cantidad de agua líquida en la capa difusora, el área efectiva de la membrana decae, resultando por tanto en un aumento de la densidad de corriente y por tanto en una caída en el voltaje (Ver Fig. 4).

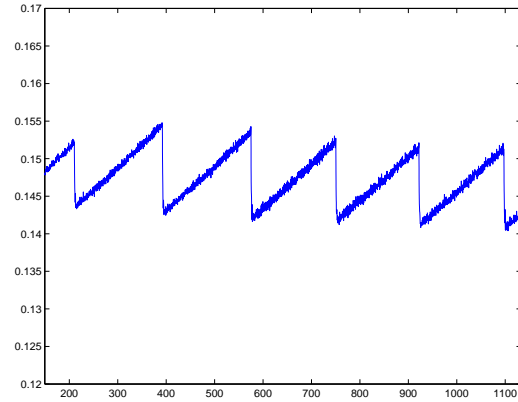


Figura 4: Evolución de la densidad de corriente durante una serie de purgas del ánodo

Así, se propone la ecuación que sigue para modelar todos los aspectos antes comentados.

$$V_{fc} = \underbrace{x_1}_{v_0} + \underbrace{x_2(T_{st} - T_{st}^0)}_{\Delta V_{fc}/\Delta T_{st}} + \underbrace{x_3 \left(0,5 \ln(p_{O_2,ca}) + \ln(p_{H_2}) \right)}_{\Delta V_{fc}/\Delta p} - \underbrace{x_4 \left(1 - \exp(-i/x_5) \right)}_{v_{act}} - \underbrace{x_6 \cdot i}_{v_{ohm}} - \underbrace{x_7 \cdot i^{(1+x_8)}}_{v_{conc}} \quad (14)$$

$$i = I_{st}/A_{fc} = I_{st}/\underbrace{A_{fc}^0(1 - \alpha_l m_{l,an})}_{A_{fc \text{ efectiva}}} \quad (15)$$

Las suposiciones que se han hecho para obtener esta ecuación son:

- Desacoplamiento de los efectos de la variación de la temperatura y de las presiones parciales de hidrógeno y oxígeno.
- Variaciones lineales con respecto a una temperatura nominal T_{st}^0 , lo que no quita generalidad a la ecuación puesto que se comprueba experimentalmente que este rango lineal cubre todo el rango práctico de operación.

- Variaciones logarítmicas con respecto a las presiones parciales de los reactivos, aunque se puede suponer que la contribución es lineal en un rango de operación entorno a un punto nominal $p_{O_2,ca}^0$ y $p_{H_2,an}^0$, que incluye todo el rango normal de operación. Esto facilita por una parte la búsqueda experimental de $\Delta V_{fc}/\Delta p_{O_2,ca}$ y por otra simplifica el algoritmo algebraico de obtención de los parámetros. La caída logarítmica se da sobre todo en las etapas de arranque y apagado de la pila, donde las presiones parciales se alejan mucho de dicho rango de operación normal. (Ver figura 5).

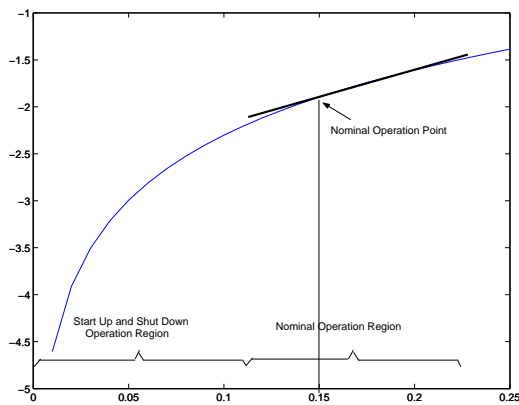


Figura 5: Tratamiento matemático de la presión de oxígeno en las diferentes zonas de trabajo de la pila de combustible

Para el cálculo de los parámetros x_1, \dots, x_8 , se ha desarrollado el algoritmo algebraico que abajo se muestra. Son necesarios 4 puntos característicos de la forma de la curva de polarización, esto es, (p_{ji}, p_{jv}) para $j = 1 \dots 4$, además de los valores de variación del voltaje con la temperatura y la presión parcial de oxígeno $\Delta V_{fc}/\Delta T_{st}$ y $\Delta V_{fc}/\Delta p_{O_2,ca}$, en un punto de funcionamiento nominal de la pila de combustible. Dichos 4 puntos han de ser elegidos de manera que dividan en 3 partes a la curva de polarización, cada una correspondiente a la zona en que dominan las pérdidas descritas al inicio de este apartado. La presión de hidrógeno se supone constante a efectos de cálculo del algoritmo, ya que su contribución viene dada físicamente como proporcional a la del oxígeno. Así, y tras escalar los valores experimentales según $\Delta V_{fc}/\Delta T_{st}$ y $\Delta V_{fc}/\Delta p_{O_2,ca}$, se obtiene una curva de polarización a una temperatura (308 K) y presiones parciales de oxígeno (0.16 bar) e hidrógeno (1.250 bar) fijas. En Fig. 6 se muestra un ejemplo de elección de los 4 puntos necesarios para el algoritmo en dicha curva. Comparando la forma de los datos de reales y la de los datos escalados

de Fig. 6, se observa además la alineación de datos que se produce tras este escalado. Esto es una comprobación experimental de las suposiciones de linealidad y desacoplamiento de los efectos de la temperatura y las presiones parciales de los reactivos antes explicadas. El parámetro α_l , por último, es

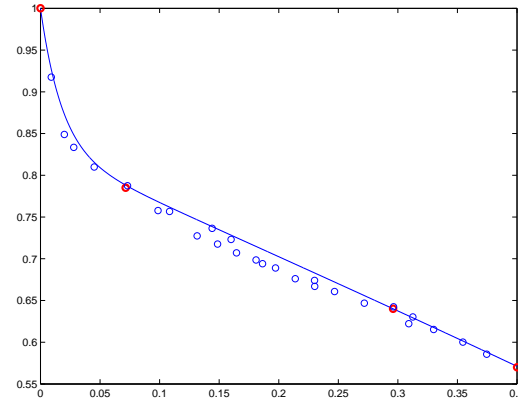


Figura 6: Elección de puntos para el algoritmo de identificación de parámetros de la curva de polarización

obtenido a partir del valor experimental de caída en el voltaje a medida que se va acumulando agua en el ánodo. Ante la imposibilidad de colocar un sensor en el interior del ánodo que mida la cantidad de agua allí condensada, se hace necesario el modelo para obtener el valor simulado de cantidad de agua líquida que corresponde con dicha caída de voltaje.

5. VALIDACIÓN DEL MODELO

Una vez planteadas las ecuaciones del modelo, hecha su implementación en MATLAB Simulink y particularizados los parámetros físicos y experimentales de dicho modelo para la pila de 1.2 kW de Ballard, el siguiente paso es la validación según datos reales tomados durante experimentos. De esta manera, se dispone de datos reales para las entradas del modelo, así como de datos de las salidas, que serán el voltaje de la pila de combustible y su temperatura. Lo que se hace es simular el modelo con esas entradas, para después comparar las salidas simuladas con las reales. Se plantean cuatro experimentos diferentes entonces: fase de arranque, carga constante para ver el efecto del encharcamiento, carga variable para comparar los transitorios y experimento de larga duración con carga variable para validar la parte del modelo dedicada a la temperatura.

5.1. Etapa de arranque

Esta etapa consiste en el periodo de tiempo desde que la pila de combustible está apagada, esto es, compresor y ventilador de aire están apagados y la válvula de hidrógeno de entrada se encuentra cerrada, hasta que se alcanzan condiciones nominales de funcionamiento. Esto ocurre cuando el voltaje que suministra la pila de combustible llega a un valor nominal estacionario. Al iniciarse el arranque, compresor y ventilador de refrigeración comienzan a funcionar, a la vez que se abre la válvula de hidrógeno de entrada. El voltaje de la pila de combustible va entonces incrementándose a medida que los reactivos van primero llenando los canales de las células de combustible y después reaccionando entre ellos. La evolución de las salidas reales del sistema pueden verse en Fig. 7, así como las salidas simuladas de voltaje y temperatura. Puede verse que la simulación predice un voltaje sensiblemente mayor que el real. Esto es debido a la dificultad de modelar la dinámica de apertura de la válvula de hidrógeno de entrada, que en el modelo se considera constante durante todo el llenado, no siendo así en la realidad, donde ocurren pequeños saltos de presión. Aún así, el resultado general puede considerarse de interés para modelar esta etapa de funcionamiento.

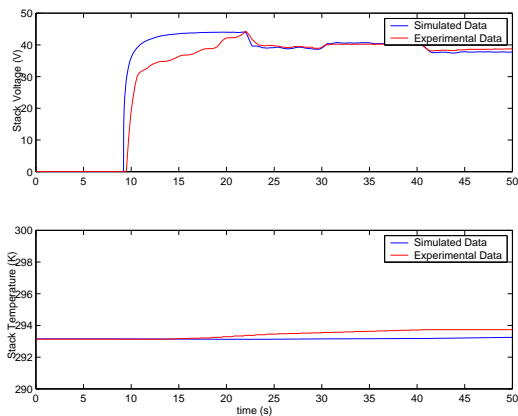


Figura 7: Variables de salida del modelo durante el arranque

5.2. Carga constante: efecto del encharcamiento

Como ya se ha descrito, durante el funcionamiento normal de la pila de combustible, se condensa y acumula agua líquida en el ánodo disminuyendo el rendimiento de las células y siendo por tanto necesarias purgas periódicas, mediante la apertura de una válvula de salida del ánodo para tales efectos. Los datos experimentales que se presentan fueron

tomados manteniendo la pila de combustible funcionando a carga constante durante un periodo de tiempo, para observar claramente el efecto del encharcamiento y la recuperación del voltaje tras las purgas (Ver Fig. 8 y Fig. 9).

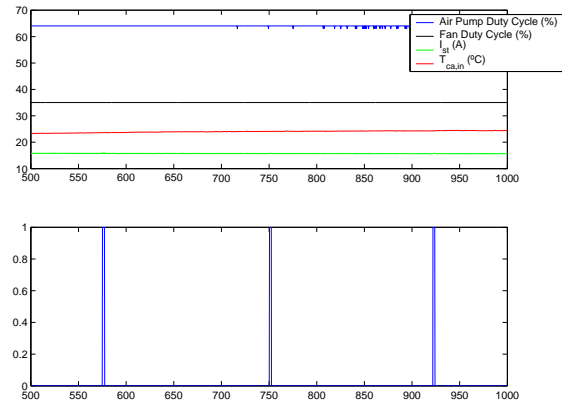


Figura 8: Variables de entrada en la simulación de los efectos del 'flooding'

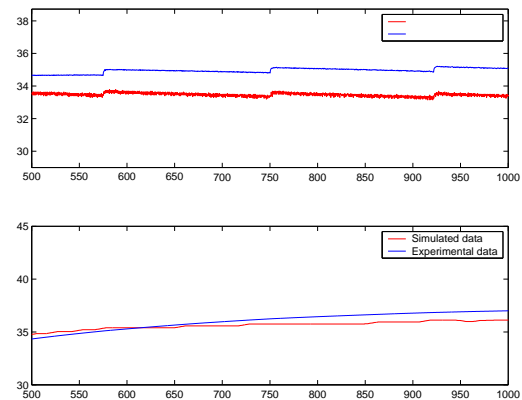


Figura 9: Variables de salida en la simulación de los efectos del 'flooding'

5.3. Carga variable: efectos transitorios

Durante grandes cambios de carga, los efectos transitorios pueden ser analizados. Estos transitorios son consecuencia de la dinámica de los fluidos. De hecho, cuando la demanda de energía aumenta, el equipo que suministra aire al sistema necesita un periodo de tiempo para poder proveer la nueva cantidad de aire requerido. Durante este tiempo, la presión parcial del oxígeno tiende a decrecer y por tanto el voltaje de la pila también. Estos efectos pueden verse en la figura 10.

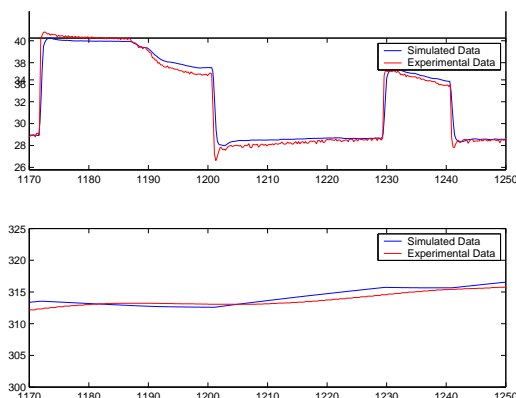


Figura 10: Variables de salida durante la simulación de los efectos transitorios

5.4. Carga Variable: temperatura

Como los efectos térmicos son mucho mas lentos que los otros, es necesaria una simulación en un gran periodo de tiempo para poder validar el bloque que modela la dinámica de la temperatura de la pila. En la figura 11 observamos la temperatura simulada sigue muy bien a la real.

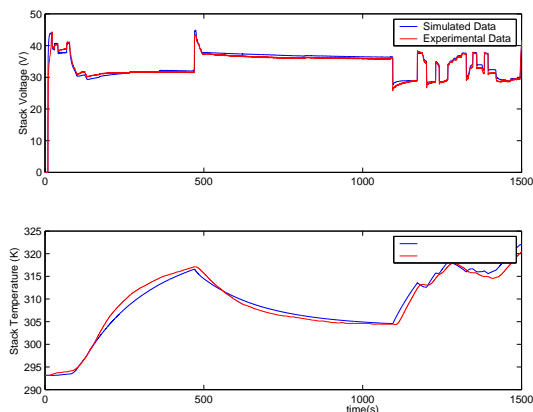


Figura 11: Variables de salida en simulación de los efectos térmicos

6. CONCLUSIONES

En este trabajo se ha desarrollado y validado un modelo dinámico para pilas de combustible tipo PEM. El modelo parte de otros existentes en la literatura y se extiende con la consideración de fenómenos de interés como son el encharcamiento o *flooding* y la consideración de los efectos térmicos. También se presenta la forma de obtener la curva de polarización de forma experimental que puede ser usado en línea.

La principal utilidad de este modelo reside en su

uso para el diseño y ensayo de algoritmos de control para la pila, facilitando el desarrollo previo a su implantación en el sistema real.

7. AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia en el marco del proyecto DPI2004-07444-C04-01 y por el Instituto Nacional de Técnica Aeroespacial.

Referencias

- [1] S. Dutta, S. Shimpalee, J.W. Van Zee, 2001. "Numerical prediction of mass-exchange between cathode and anode channels in a PEM fuel cell". *International Journal of Heat and Mass Transfer*, **44**, pp. 2029–2042.
- [2] J.H. Nam, M. Kaviany, 2003. "Effective diffusivity and water-saturation distribution in single- and two-layer PEMFC diffusion medium". *International Journal of Heat and Mass Transfer*, **46**, pp. 4595–4611.
- [3] D.A. McKay, W.T. Ott, A.G. Stefanopoulou, 2005. "Modeling, parameter identification, and validation of reactant and water dynamics for a fuel cell stack". Proceedings of IMECE 2005, ASME International Mechanical Engineering Congress & Exposition.
- [4] E.A. Müller, A.G. Stefanopoulou, 2005. "Analysis, modeling, and validation for the thermal dynamics of a polymer electrolyte membrane fuel cell system". Proceedings of FUELCELL 2005, Third International Conference on Fuel Cell Science, Engineering and Technology.
- [5] X. Yu, B. Zhou, A. Sobiesiak, 2005. "Water and thermal management for Ballard PEM fuel cell stack". *Journal of Power Sources*, **147**, pp. 184–195.
- [6] J.T. Pukrushpan, H. Peng, A.G. Stefanopoulou, 2004. "Modeling and analysis of fuel cell reactant flow for automotive application". *Journal of Dynamic Systems, Measurement and Control*, **126** (1), pp. 14–25.
- [7] Zemansky, Dittman. Heat and Thermodynamics, McGraw Hill, copyright dates from 1937 to 1981
- [8] J.C. Amphlett, R.M. Baumert, R.F. Mann, B.A. Peppley, P.R. Roberge, 1995. "Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell". *Journal of Electrochemical Society*, **v.142, n.1**, pp. 9–15.

CONTROL DE UNA GRÚA PÓRTICO MEDIANTE REDES NEURONALES

Alejandro Fernández Villaverde
Dpto. de Ingeniería de Sistemas y Automática
E.T.S.I.I. - Universidade de Vigo
Campus Universitario As Lagoas-Marcosende. 36200 Vigo
e-mail: afernandez@uvigo.es

Matilde Santos Peñas
Dpto. de Arquitectura de Computadores y Automática
Facultad de Informática - Universidad Complutense de Madrid
e-mail: msantos@dacya.ucm.es

Resumen

El control de una grúa pórtico representa un desafío para los ingenieros de control debido a que es un sistema complejo, multivariable y no lineal. A lo largo de las últimas décadas se han propuesto diferentes enfoques para el tratamiento de este problema, habitualmente aplicando técnicas clásicas al control. En este artículo se presenta una solución basada en una de las técnicas de la Inteligencia Artificial, las redes neuronales, para la identificación y el control basado en linealización por realimentación del sistema. Los resultados de simulación prueban la eficiencia de esta estrategia para controlar este sistema.

Palabras Clave: Grúa Pórtico, Inteligencia Artificial, Linealización por Realimentación, Redes Neuronales

1 INTRODUCCIÓN

El objetivo de este trabajo es realizar la identificación y el control de una grúa pórtico de tres grados de libertad mediante redes neuronales.

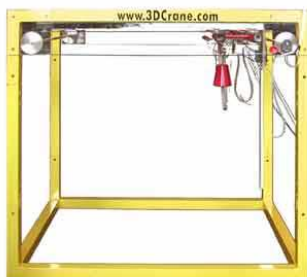


Figura 1. Vista general de la grúa

Una grúa pórtico o puente grúa (*gantry crane* o *overhead crane*) es un dispositivo ampliamente

usado en la construcción, transporte (puertos) e industria. Al mismo tiempo, se trata de un sistema mecánico de elevada complejidad, ya que presenta un comportamiento no lineal y de naturaleza multivariable. En consecuencia, muchas de las técnicas “clásicas” de control no pueden ser aplicadas. Además, la dinámica del sistema propicia la aparición de oscilaciones indeseadas de la carga que son difíciles de atenuar. Por ello las grúas han sido objeto de abundantes artículos y estudios en el último medio siglo.

Para la realización de este trabajo se ha partido del modelo 3DCrane de la casa polaca Inteco (ver figura 1). Esta grúa, cuyas dimensiones son las de un cubo de aproximadamente 1.5 metros de arista, está diseñada para un uso didáctico o de investigación. Consta de una viga móvil que se desplaza en una dirección x sobre unos raíles, y de un carro que se desplaza a lo largo de la viga, en una dirección y perpendicular a x . Del carro cuelga un cable que sostiene la carga, pudiendo ésta ser izada o bajada a lo largo del eje z . De esta forma, es posible posicionar la carga en cualquier punto del espacio de trabajo xyz de la grúa (3D). Los ejes son accionados por motores de corriente continua, y existen encoders que miden la posición (x , y , z) y los ángulos de desviación de la carga respecto a la vertical. En la figura 2 se puede ver con más detalle la configuración de la grúa.

Para el control del sistema se utilizarán redes neuronales. Esta técnica de la IA se ha mostrado muy versátil para estas tareas. Para ello, en la Sección 2 se selecciona una determinada arquitectura neuronal y se expone el algoritmo utilizado para entrenar la red para que cumpla lo mejor posible con las especificaciones de diseño así como el algoritmo de control. En la siguiente sección se especifica su aplicación como sistema de control de la grúa pórtico. Los resultados obtenidos mediante la

simulación se muestran en la sección 4. Con las conclusiones y futuros trabajos finaliza esta contribución.

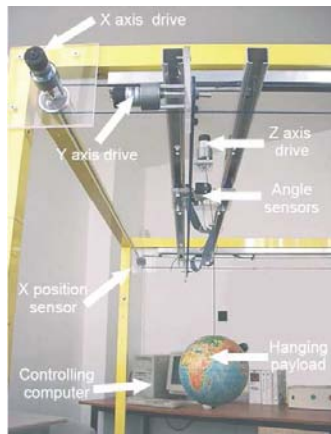


Figura 2. Detalle de la grúa-pórtico

2. RED NEURONAL UTILIZADA

2.1 TIPO DE RED

En esta aplicación las redes neuronales se usan con el propósito de realizar una linealización por realimentación, es decir, cancelar las no-linealidades de la planta.

Al tratarse de un sistema dinámico, es necesario que las salidas de la red dependan no sólo de los valores actuales de las entradas, sino también de las entradas, salidas y/o estados anteriores. Es decir, se precisa que la red sea dinámica [4]. Este tipo de redes son más potentes, al poseer memoria. Entre sus aplicaciones más típicas se encuentran los sistemas de control [3].

El tipo de red utilizado es el *Layered Digital Dynamic Network*, descrita en [2]. Es similar al clásico perceptrón multicapa, pero con una línea de retardos. El aprendizaje o modificación de los pesos sinápticos se realiza mediante el mecanismo de retropropagación. Merece la pena detenerse brevemente a explicar qué tipo de aprendizaje o entrenamiento se ha elegido.

2.2 APRENDIZAJE

Hay diferentes métodos para entrenar la red. El más sencillo consiste en actualizar los pesos en la dirección en la que la función criterio del perceptrón desciende más rápidamente, es decir, en la dirección del gradiente negativo. Una mejora a este método consiste en añadirle un “momentum”, de forma que la red responda no sólo al valor local del gradiente, sino a la tendencia. De esta forma se consigue un aprendizaje más rápido. Aún así, estos métodos

pueden resultar demasiado lentos, por lo que en su lugar se ha utilizado la modificación conocida como algoritmo de Levenberg-Marquardt [1]. Su fundamento es el siguiente: a pesar de que la función criterio del perceptrón desciende más rápidamente a lo largo de la dirección indicada por el gradiente, esto no asegura que de esta manera se logre la convergencia más rápida. Como alternativa, se puede realizar una búsqueda a lo largo de direcciones conjugadas. Surgen así los Algoritmos de Gradiente Conjugado. Una variante de estos algoritmos son los llamados Algoritmos Quasi-Newton, en los cuales se actualizan los pesos mediante la matriz Hessiana de la función (sus segundas derivadas), evaluada en los valores actuales de los pesos. Como la matriz Hessiana es costosa de calcular, una alternativa consiste en aproximarla mediante la expresión $H = J^T \cdot J$, donde J es el jacobiano (de los errores de la red respecto a los pesos), con lo que la actualización de los pesos se realiza de la forma siguiente:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \cdot \mathbf{J} + \mu \cdot \mathbf{I}]^{-1} \cdot \mathbf{J}^T \cdot \mathbf{e} \quad (1)$$

donde \mathbf{e} es el vector de errores y μ es un parámetro que se va reduciendo a medida que se va acercando a la solución, haciendo así que este método sea más parecido al de Newton, por ser éste más rápido en las proximidades de los mínimos. Éste es el algoritmo de Levenberg-Marquardt.

Una vez entrenada la red neuronal, se puede integrar en el sistema de control de la planta, lo que se detalla en el siguiente apartado.

3. SISTEMA DE CONTROL

El control se realiza desde un PC equipado con una tarjeta PCI suministrada por el fabricante de la grúa Inteco. Esta tarjeta se comunica con la grúa a través de un interfaz de potencia, al cual se le conectan tanto los motores como los sensores. La implementación del control a nivel software (tanto para simulaciones como para el modelo real) se ha hecho en un entorno Matlab/Simulink trabajando bajo Windows. El interfaz con la planta utiliza la *Crane3D Toolbox* proporcionada por la propia Inteco, que se integra fácilmente dentro de este entorno. Para la identificación del sistema y el diseño del controlador se usó la *Neural Network Toolbox* de Matlab. Finalmente, para poder realizar control en tiempo real se precisa de la *Real Time Workshop* y el *Real Time Windows Target*.

El tipo de control realizado es, como ya se indicó, de linealización por realimentación. Este tipo de control consiste en lo siguiente: sea un sistema no-lineal que

se puede modelar, empleando la notación usual en teoría de sistemas en tiempo discreto, como:

$$y(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] + g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] \cdot u(k+1) \quad (2)$$

donde $d \geq 2$

Si se puede realizar una identificación del modelo, es decir, encontrar una red neuronal que aproxime las funciones f , g , entonces se puede obtener la entrada de control como:

$$u(k+1) = \frac{y(k+d) - f[\dots]}{g[\dots]} \quad (3)$$

De esta forma, el controlador se deriva directamente del modelo neuronal de la planta.

Hay que señalar que la linealización por realimentación puede ser realizada sin usar redes neuronales. Sin embargo, su uso se justifica por su relativa simplicidad, ya que los cálculos necesarios para linealizar el sistema de la forma clásica son farragosos, incluyendo el manejo de matrices grandes y el cálculo de derivadas de las salidas. Además, la expresión analítica del controlador obtenido suele ser complicada y por lo tanto difícil de implementar.

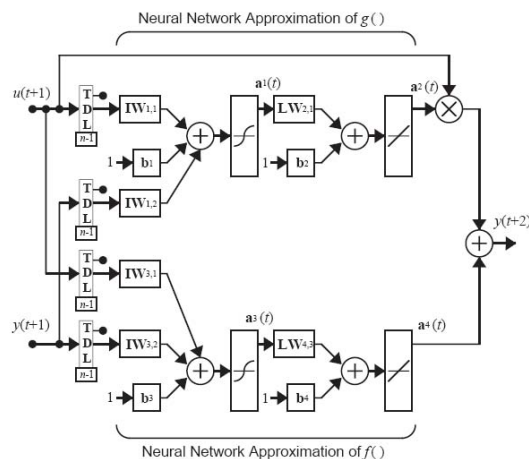


Figura 3. Aproximación de f , g

En nuestro caso se usó el bloque NARMA-L2 (Nonlinear AutoRegressive Moving Array) de la Neural Network Toolbox de Matlab. Este bloque realiza la identificación del sistema y calcula la señal de control necesaria. Su principal ventaja es su reducida carga computacional, factor importante ya que el entrenamiento de una red neuronal exige un

elevado uso de la CPU, y, en caso de ser demasiado compleja, puede llegar a ser imposible llevar a cabo un aprendizaje adecuado. El esquema de cómo la red neuronal realiza la aproximación de las funciones f , g se muestra en la figura 3, y un esquema de control en la figura 4. Estas figuras siguen la notación adoptada por Matlab y fueron obtenidas de [5].

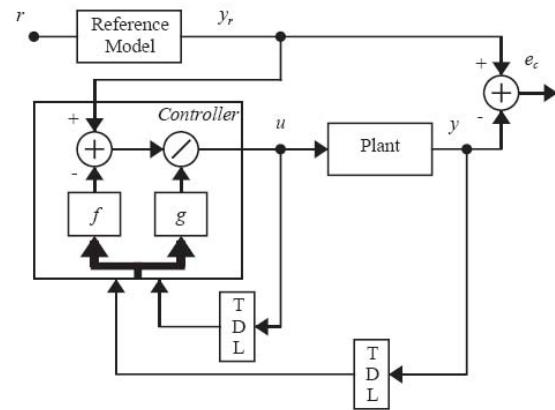


Figura 4. Sistema de Control

En el diseño de la red para esta aplicación se ha optado por una configuración con una capa oculta con 4 neuronas, cuyas funciones de activación son sigmoides. La capa de salida representa las salidas del sistema y sus funciones de transferencia son lineales. Para cada capa de salida y de entrada se utilizaron dos entradas y salidas, respectivamente, retardadas. De esta forma se puede llegar a aproximar un sistema mecánico de segundo orden, como es la grúa.

El aprendizaje se realiza, tal como se justificó en la sección anterior, mediante el algoritmo de Levenberg-Marquardt, que en Matlab está implementado en la función `trainlm`.

4 EXPERIMENTOS

En primer lugar se simuló el comportamiento del sistema utilizando un modelo no lineal de la planta. De esta forma se ajustaron los controladores hasta obtener un comportamiento satisfactorio de la planta en lazo cerrado. Para ello, se modeló la grúa utilizando la formulación Lagrangiana. El sistema resultante consta de 3 entradas (los 3 motores), 5 salidas, y 10 estados. Este modelo se incluyó en un diagrama de Simulink, al cual se añadieron los bloques de NARMA-L2 (uno por cada eje de la grúa).

A efectos de la identificación del sistema, la grúa presenta como entradas las fuerzas aplicadas, (tres, una por cada eje), y como salidas las posiciones

respectivas. Por lo tanto, cada bloque NARMA-L2 que implementa una red se encarga de la identificación y control de un eje de la grúa. Las funciones f y g obtenidas de la identificación permiten obtener la acción de control mediante (3). La red lo que hace es aproximar de forma lineal el sistema y a partir de esas aproximaciones se obtiene directamente la señal de control. Además este tipo de control suele tener problemas de robustez si el modelo no es muy exacto, lo que se evita si es la

misma red la que identifica y obtiene el control con la estrategia indicada de linealización por realimentación. Esta identificación neuronal evita los cálculos derivados de una linealización clásica del modelo.

La identificación de cada eje se realizó en base a 2000 muestras, con un periodo de muestreo de 0.05 segundos.

Control con RR.NN. de la Grúa Simulada

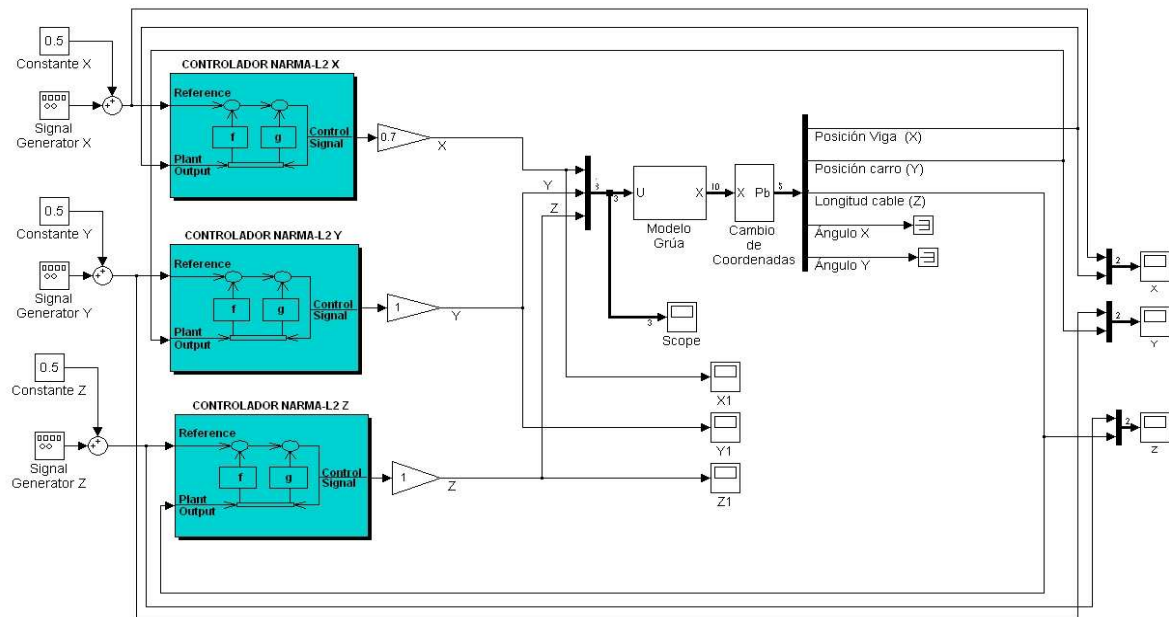


Figura 5. Diagrama de control en Simulink

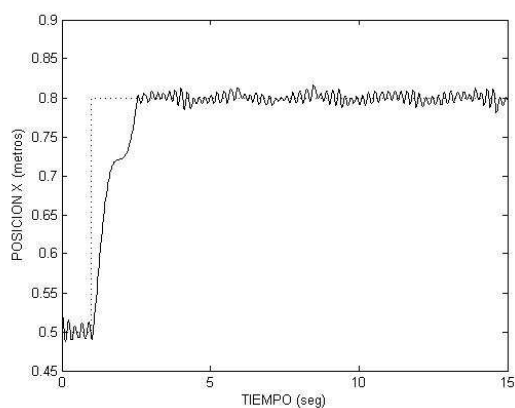


Figura 6. Respuesta a escalón en la viga (eje X)

El diagrama de control utilizado se muestra en la figura 5. Se realizaron pruebas para medir tanto la respuesta a entrada escalón como el seguimiento de una señal senoidal. En las figuras 6-12 se muestran las diferentes respuestas a estas entradas, así como las señales de control.

Nótese que se ha modificado la señal de control del eje x , multiplicándola por 0.7. Esto se ha hecho con el fin de ilustrar las diferentes posibilidades de ajuste de los controladores. Así, en el eje Y, cuya señal se ha dejado intacta, se tiene una respuesta más rápida, aunque se llega a un valor ligeramente superior al de la consigna. Por el contrario, en el eje X, si bien con algo más de lentitud, se consigue exactamente el valor de consigna, sin tener sobreoscilación. Ambas respuestas son aceptables y deberá escogerse una u otra en función de la aplicación.

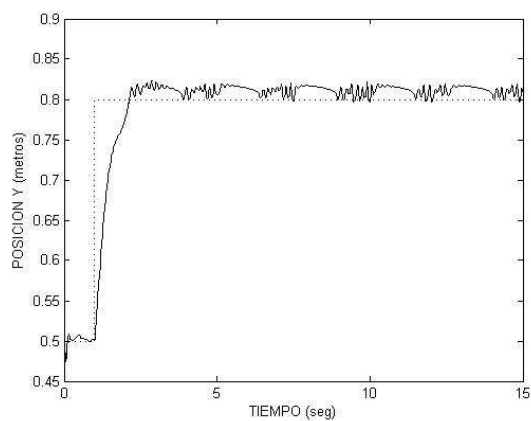


Figura 7. Respuesta a escalón en el carro (eje Y)

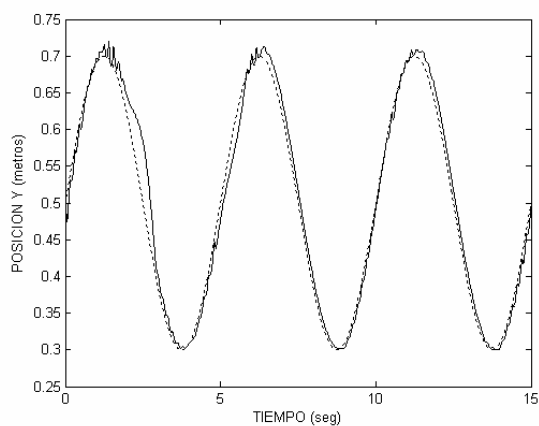


Figura 10. Respuesta a senoide en el carro (eje Y)

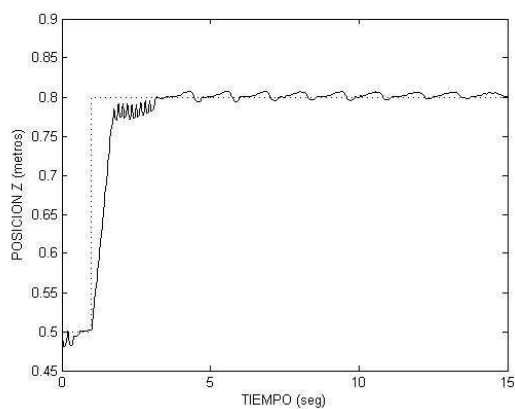


Figura 8. Respuesta a escalón en el cable (eje Z)

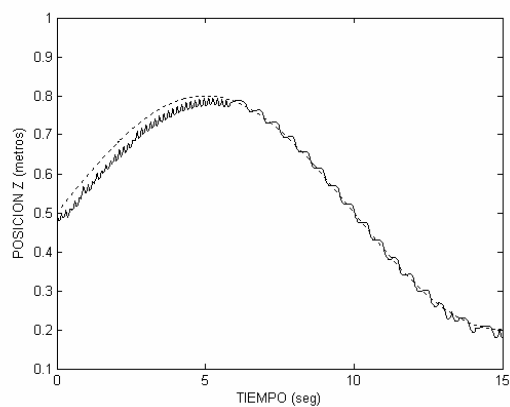


Figura 11. Respuesta a senoide en el cable (eje Z)

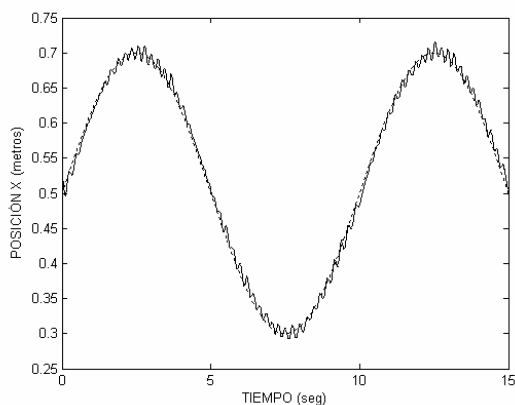


Figura 9. Respuesta a senoide en la viga (eje X)

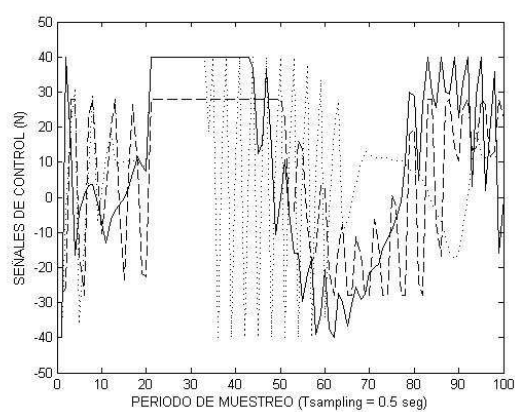


Figura 12. Señales de control para respuesta a escalón (línea: X, punto: Y, sólido: Z)

5 CONCLUSIONES

En este documento se ha presentado una aplicación de las redes neuronales artificiales al control de una grúa pórtico. Se comprueba que, pese a la complejidad que presenta la planta a controlar (ya que es no lineal y multivariable), la aplicación de estas técnicas es sistemática y permite obtener buenos resultados. La estructura de la red fue escogida de forma que se minimizara la carga computacional, facilitando la viabilidad del diseño. Como estructura del controlador se utilizó la de linealización por realimentación, por lo que, una vez identificada la planta, el cálculo del controlador fue inmediato. Este diseño del controlador fue comprobado mediante simulación, en la que se ajustaron las ganancias para obtener el control más ajustado posible.

Las futuras líneas para continuar este trabajo podrían partir de un estudio más pormenorizado de las oscilaciones angulares producidas en la carga de la grúa, y del diseño de una estrategia para minimizarlas. Asimismo, se debería probar los resultados de las simulaciones en la planta real, lo que no pudo ser hecho hasta la fecha debido a problemas con el hardware.

Agradecimientos

Este trabajo se inscribe en el proyecto DPI2004-07670-C02-01, el cual ha sido financiado parcialmente por el Ministerio de Ciencia y Tecnología.

Referencias

- [1] Ampazis, N. and Perantonis, S.J., (2000) "Lavenberg-Marquardt algorithm with adaptive momentum for the efficient training of feedforward networks", *Proc. of the IEEE-ENNS Int. Joint Conf. on Neural Networks*, vol. 1, 24-27 July, pp. 126-131.
- [2] De Jesús, O., Hagan, M., (2001) "Backpropagation Through Time for a General Class of Recurrent Network", *Proceedings of the International Joint Conference on Neural Networks*, Washington, DC, pp. 2638-2642.
- [3] De Jesús, O., Demuth, H., Hagan, M., (2002) "An Introduction to the Use of Neural Networks in Control Systems", *International Journal of Robust and Nonlinear Control*, Vol. 12, No. 11, pp. 959-985.
- [4] Delgado, A., Kambhampati, C., Warwick, K., (1995) "Dynamic recurrent neural network for system identification and control", *IEE Proceedings - Control Theory and Applications*, Volume 142, Issue 4, p. 307-314.
- [5] Demuth, H., Beale, M., Hagan, M., (2006) "Neural Network Toolbox User's Guide" – The Mathworks, Natick, MA.

Aplicación de tiempo real para el control de campos de heliostatos

Jose González, Luis Yebra, Antonio Valverde, Manuel Berenguel, Mario Peralta
 CIEMAT-Plataforma Solar de Almería (PSA) Dpto. Lenguajes y Computación
 Carretera Senés s/n Universidad de Almería
 04200 Tabernas, Almería Ctra. Sacramento s/n
 {jose.gonzalez, luis.yebra, antonio.valverde}@psa.es 04120 Almería
 beren@ual.es, mpc@alboran.ual.es

Resumen

Las actividades realizadas en plantas solares suelen producir grandes concentraciones de radiación solar que pueden alcanzar temperaturas muy elevadas. Por ello, errores que se puedan producir tanto en el resultado de los cálculos de las posiciones donde deben incidir las concentraciones como en el tiempo de respuesta a la hora de indicar estas posiciones pueden ocasionar daños en las personas o en los componentes irradiados. El cumplimiento de requisitos temporales y de seguridad hacen sea conveniente aplicar técnicas de Tiempo Real a aplicaciones sobre plantas solares. En este artículo se describe como se está implementando una aplicación usada en una planta solar de manera que cumpla tanto las restricciones temporales como de seguridad.

Palabras clave: Sistemas de tiempo real distribuido, plantas termosolares, sistemas de receptor central

1. Introducción

Para producir energía eléctrica, la energía solar se puede convertir en energía calorífica concentrando los rayos solares sobre una superficie. Esta concentración puede producir sobre las superficies irradiadas temperaturas muy altas. Entre los sistemas basados en métodos térmicos destacan los llamados de receptor central.

El grupo de investigación “Automática, Electrónica y Robótica” (AER, código TEP-197 del Plan Andaluz de Investigación) de la Universidad de Almería (UAL) está realizando un proyecto de investigación titulado “Desarrollo de sistemas y herramientas de control para plantas termosolares” (bajo el convenio PSA-UAL-AER). Entre los objetivos del proyecto se encuentra el desarrollo de un nuevo Sistema de Control de Campos de Heliostatos (SCCH) a implantar en los campos de la Plataforma Solar de Almería (PSA-CIEMAT). En la figura 1 se pueden ver las instalaciones de sistemas de receptor central que dispone la PSA: CESA-I y CRS. Este nuevo SCCH sustituirá a una

aplicación existente situada en un μ VAX programada en el lenguaje FORTRAN.



Figura 1: Sistemas de Receptor Central en la PSA. CESA-I en la parte superior y CRS en la parte inferior.

Aparte de la aplicación para el control de campos de heliostatos, el grupo ha realizado un proyecto dedicado a la implementación de una aplicación para el control del receptor y sistema de potencia [1], el cual ha sido desarrollado usando LabVIEW de *National Instrument* junto con el módulo *Data-logging and Supervisory Control Module*. También se ha desarrollado una aplicación para el ajuste automático de offset (desviaciones estáticas) usando técnicas de visión artificial [2].

En este artículo se va a describir la aplicación de tiempo real para el control de campos de heliostatos. También se describirá brevemente la aplicación para el ajuste automático de offsets que hace uso de la aplicación que se está desarrollando. En las siguientes secciones se muestra una descripción de los sistemas de receptor central y del elemento principal a controlar (el heliostato). Posteriormente se explican el estado actual del desarrollo de la aplicación control de campos de heliostatos en plantas termosolares y, finalmente, se mostrarán algunas conclusiones y trabajos futuros que se pretenden realizar.

2. Sistemas de receptor central

Los sistemas de receptor central concentran la energía solar sobre un foco. Estos sistemas se dividen en dos subsistemas: por una parte el campo de heliostatos (encargada de concentrar la radiación solar), y por otra el receptor (como foco donde se concentra la radiación) junto al sistema de potencia (cuya misión es la generación de energía). El campo de heliostatos está compuesto por un conjunto de espejos curvos (heliostatos) que reflejarán y concentrarán la radiación solar que incidente sobre el campo en unos puntos determinados dentro del receptor [5]. Existen una serie de factores que dificultan la concentración constante dentro del receptor, necesaria para un buen funcionamiento del sistema. Entre estos factores están la dependencia temporal de los sistemas que usan energía solar (la intensidad de la radiación varía a lo largo del día) y las posibles perturbaciones (por ejemplo, por presencia de nubes).

2.1. El heliostato

Los heliostatos poseen dos ejes (elevación y azimut) con sendos grados de libertad que permitirán orientarlos para concentrar la radiación solar. Cada heliostato del campo dispone de un control local que se encarga de controlar los motores de estos ejes para posicionar al heliostato en las consignas deseadas. Básicamente, las operaciones que realiza un heliostato son posicionarse en una orientación fija o realizar el seguimiento de una coordenada.

El seguimiento continuo a una coordenada hace que el rayo reflejado por el heliostato quede incidiendo constantemente sobre un punto del espacio, a pesar del movimiento del sol. Esto permitirá la concentración en un punto deseado del espacio de la radiación solar que incide sobre el campo de heliostatos. El seguimiento del punto supone el cálculo de los valores de consigna de elevación y azimut para el heliostato. Los valores calculados dependen de (1) la coordenada donde se quiere incidir, (2) de la posición del heliostato en el campo (respecto a la torre) y (3) de la fecha y hora con la que se puede obtener la posición del sol en el cielo. Aunque las coordenadas donde se quiere hacer incidir el rayo reflejado no cambien es necesario recalcular periódicamente las consignas de elevación y azimut puesto que el sol cambia su posición en el cielo constantemente.

3. Aplicación para el control del campo de heliostatos

Los objetivos que se pretenden cumplir con la aplicación que se está desarrollando son:

- La aplicación tiene que ser *genérica* dentro del control de campos de heliostatos. El uso de la aplicación en otros campos de heliostatos o la modificación del campo donde se encuentre instalado no debe suponer un gran esfuerzo. Para ello, las alteraciones en la aplicación no debería exceder, de forma general, de la modificación de la base de datos con la información sobre el campo usada por la aplicación y de la creación o modificación de una clase en C++ con el protocolo de comunicaciones usado en el campo.
- La aplicación se dividirá en tres partes: la interfaz para el operador, el subsistema de control del campo y el subsistema de comunicaciones. La interfaz para el operador se encarga de suministrar los comandos y acciones indicadas por el operador al subsistema de control e informar al operador sobre el estado en el que se encuentra el campo. Por su parte, el subsistema de control del campo proporcionará una Interfaz Lógica de Tiempo Real hacia los otros sistemas como pueden ser la interfaz del operador, el sistema para el control del receptor o la aplicación para el ajuste automático de offsets. Finalmente, el subsistema de comunicaciones se encargará de interactuar con el campo de heliostatos transmitiendo las tramas de comunicación y recibiendo las respuestas de los heliostatos.
- Se intentará modularizar la aplicación lo más posible con la idea de que ésta pueda ser fácilmente *distribuida* en varios computadores. Esta distribución ayudaría a hacer la aplicación más escalable en el caso de que se usase para controlar campos de heliostatos muy grandes.
- La aplicación será *independiente de la tecnología de comunicaciones* utilizada por los heliostatos. Para empezar, se debe poder comunicar con los campos de heliostatos de la PSA donde la comunicación se realiza tanto a través de cable como por medios inalámbricos, usando cada uno un protocolo distinto.

Se han definido los siguientes requisitos temporales para el sistema:

- El subsistema de control debe comunicarse con cada heliostato al menos una vez cada 4 segundos. Por tanto, debe ser capaz, en el peor de los casos, de realizar los cálculos de consignas para todos los heliostatos, comunicarlos a los controles locales que correspondan y esperar la respuesta. Puesto que el número de heliostatos del campo y la capacidad

de computación de la que dispone el control central pueden variar de un campo a otro, es posible que sea necesaria la distribución de la aplicación para que estos tiempos se cumplan.

- Se deben adquirir una serie de temperaturas y señales cada segundo. Los valores de estas temperaturas, cuando rebasan ciertos límites, y de las señales pueden generar una alarma y desenfocar todos o parte de los heliostatos cuyo rayo reflejado apunte al receptor.
- Todos los comandos ejecutados por los heliostatos tendrán un tiempo máximo para su finalización.

La aplicación se ejecutará sobre el sistema operativo de tiempo real LynxOS. Para la creación de los distintos módulos software se está haciendo uso de las librerías ACE ([6], [7]) y utilizando sus capacidades para tiempo real. Estas librerías ayudan a conseguir independencia del API proporcionado por el Sistema Operativo por lo que módulos completos pueden ser reutilizados al ser recompilados en distintas plataformas. Se está haciendo uso también de una implementación de RT-CORBA basada en componentes ACE llamada TAO (The ACE ORB) [9]. En lo que respecta a la interfaz gráfica para el operador, se están usando el entorno gráfico X-Window y el estándar Motif.

El uso de objetos RT-CORBA permitirá la distribución de la aplicación a la vez que ofrece un interfaz lógico a otras aplicaciones como el sistema para la corrección automática de offsets. En la figura 2 se muestra la relación del SCCH con otros sistemas. La elección de la distribución de CORBA TAO se debe a su amplio uso tanto industrial como científico y a que, puesto que está basada en componentes ACE, puede usarse en gran número de plataformas, entre las que se encuentran Microsoft Windows, Linux y LynxOS, que son las usadas principalmente en PSA.

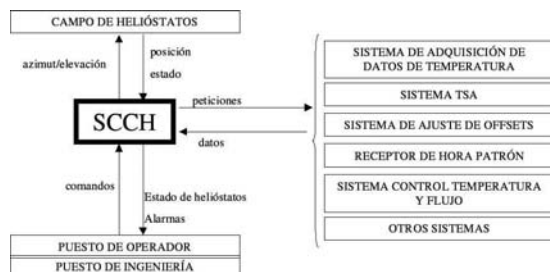


Figura 2: Diagrama de contexto

Como se ha comentado, la aplicación se divide principalmente en tres módulos: la interfaz, el subsistema de control y el subsistema de comunicaciones. Estos pueden estar separados o puede ejecu-

tarse en diferentes computadores. Es más, es posible disponer de varios procesos interfaz (en el mismo o en distintos computadores) que usen los servicios de un mismo subsistema de control. La figura 3 muestra un esquema de estos tres módulos. En ella aparecen 16 líneas que comunican el subsistema de comunicaciones con el campo de heliostatos. También aparecen 300 conexiones entre el subsistema de control y el subsistema de comunicaciones. Estas 16 líneas se corresponden a las líneas de comunicaciones usadas en el campo CESA-I y las 300 conexiones a los 300 heliostatos del campo.

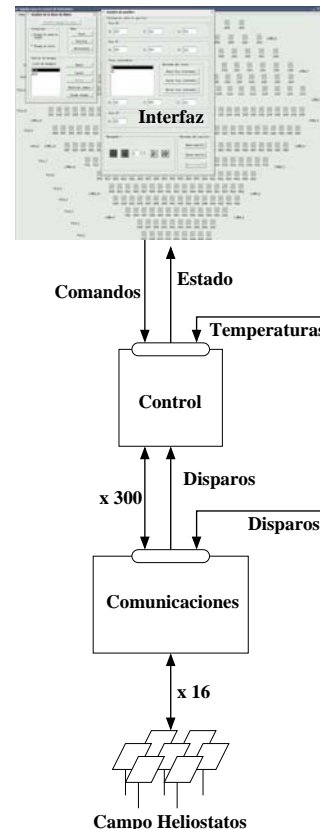


Figura 3: Esquema general de la aplicación para el campo CESA-I de la PSA

A continuación se pasa a describir cada una de las partes de la aplicación junto con la aplicación desarrollada para el ajuste automático de offsets.

3.1. Interfaz de operador

Para la manipulación del campo de heliostatos se ha desarrollado una interfaz que será usada por un operador. Esta interfaz se encargará de gestionar los distintos tipos de ensayos disponibles para cada campo, de hacer llegar los comandos al subsistema de control y de mostrar información acerca del campo de heliostatos. En la figura 3 se puede ver, en la parte superior, la interfaz donde se apre-

cia, en el fondo, la pantalla principal mostrando el campo de heliostatos y, delante de ésta, dos ventanas que permiten gestionar ciertos parámetros para los distintos ensayos del campo.

3.2. Subsistema de control

En el subsistema de control existen varios módulos principales, de entre los cuales los objetos heliostato son los más importantes. En la figura 4 se muestra este subsistema y la organización de sus módulos. Tanto la factoría como los objetos heliostato y los módulos de disparo son objetos RT-CORBA. Inicialmente sólo existe el módulo factoría, el módulo de temperatura y el del cálculo del vector solar. La factoría creará, a través de peticiones CORBA, el resto de componentes. A continuación describimos estos módulos.

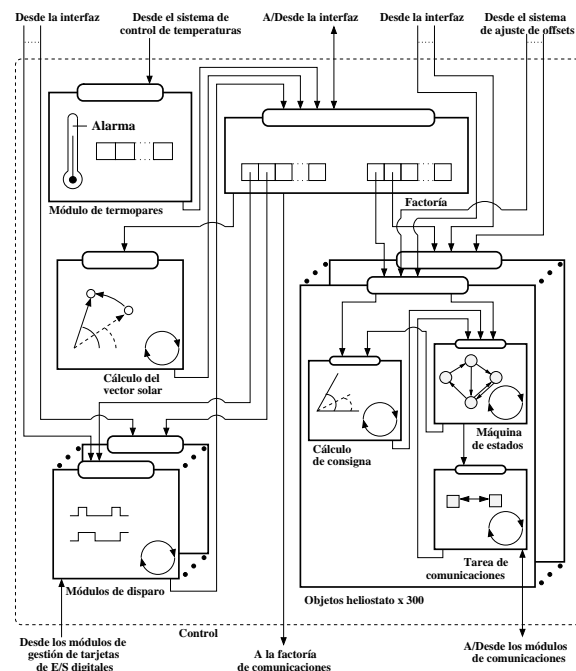


Figura 4: Subsistema de control

3.2.1. Objetos heliostato

El heliostato será el componente principal del campo y deberá ser controlado individualmente. Para ello se ha creado un objeto lógico activo con respuesta determinista que almacena la información relativa al heliostato físico y gestiona operaciones de alto nivel invocadas a través de su interfaz CORBA. Este objeto activo tendrá hasta tres tareas independientes: la primera de ellas se encarga de gestionar el estado del heliostato, la segunda tarea realiza las comunicaciones con el control local del heliostato físico y la tercera tarea recalcula la orientación del heliostato cuando se realiza el seguimiento sobre un foco.

Se ha optado por separar la manipulación del estado actual del objeto heliostato y las comunicaciones de éste debido a que las comunicaciones pueden incurrir en bloqueos ocasionados por otros heliostatos pertenecientes a una misma línea de comunicaciones compartida. Describimos ahora cada una de las tareas más detenidamente.

1. *Máquina de estados*: Esta tarea mantiene el estado actual del heliostato manipulado una máquina de estados que ha sido expresamente diseñada. Esta máquina, además de mantener el estado actual, establece las transiciones permitidas entre los estados. Las transiciones serán activadas ante la llegada de eventos que pueden generarse al realizar llamadas a la interfaz CORBA del objeto heliostato (como puede ser el indicar al heliostato que dirija el rayo reflejado hacia el receptor) o por respuestas específicas del control local del heliostato físico (como puede ser la señalización de un fallo en algún motor). En la imagen 5 se muestra el diagrama de la máquina de estados usada por los heliostatos.

Un heliostato, visto por el operador, se encontrará en un *estado lógico* (por ejemplo, en seguimiento normal sobre el receptor, abatido, en error, etc.). Estos estados lógicos son distintos a los estados de la máquina de estados (ME), pero a cada uno de los estados lógicos le corresponde un estado de la ME. Los nombres de los estados de la ME que tienen un equivalente lógico comienzan por E_. Además de éstos, la ME está compuesta por otros estados que representan la transición de un estado lógico origen a otro destino debido a la ejecución de un comando (el nombre de los estados comienzan por X_ o por T_) y, también, la transición hacia un estado lógico origen debido a la cancelación del comando (el nombre de los estados comienzan por CA_).

2. *Tarea de comunicaciones*: Esta tarea se encarga de realizar las comunicaciones con el heliostato. La tarea mantiene un mensaje que será enviado periódicamente al heliostato. Este mensaje es modificado tanto por la tarea de la ME como la tarea que realiza el cálculo de la consigna. El tiempo máximo entre dos envíos del mensaje será de 4 segundos, pudiéndose mandar un mensaje con menor tiempo respecto al anterior cuando alguna situación lo requiera. Cada vez que la tarea envía el mensaje al heliostato, espera la respuesta de éste actualizando la información de la consigna actual y el estado del heliostato, pudiendo generarse algún cambio de estado en la tarea ME.

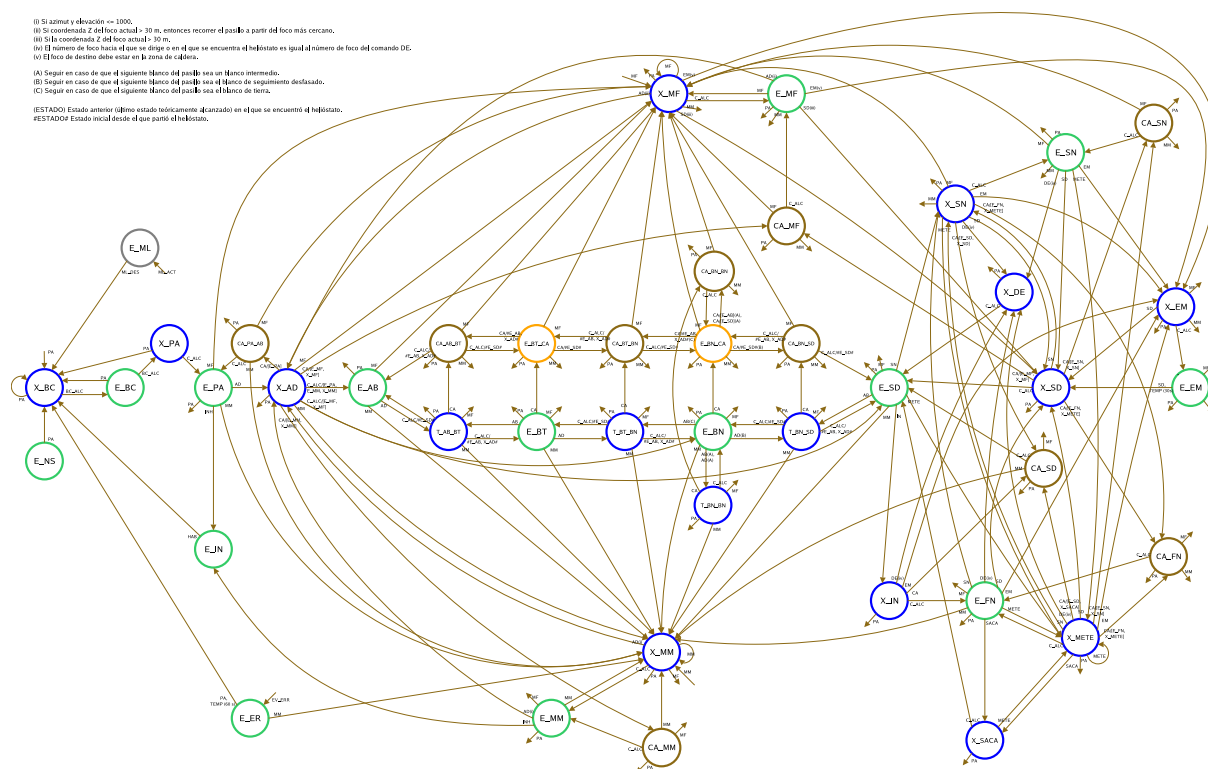


Figura 5: Máquina de estados que define el comportamiento del heliostato

3. *Cálculo de consigna:* Además de las dos tareas comentadas anteriormente que ejecuta un objeto heliostato es posible que, dependiendo del estado actual en el que se encuentre, este objeto disponga de otra tarea adicional asociada al estado actual. Cuando el heliostato se encuentra en un estado en el que se debe realizar el seguimiento de un punto (esto es, dirigir el rayo reflejado de forma que incida en una coordenada concreta) es necesario realizar una actividad consistente en recalcular periódicamente las consignas de los ejes de elevación y azimut debido al continuo movimiento del sol. Estos cálculos serán realizados por esta nueva tarea que será activada al entrar a un estado en el que se precise hacer seguimiento.

3.2.2. Cálculo del vector solar

Calcula la posición del sol a partir de la localización de la planta, la hora y la fecha actual, y actualiza los valores obtenidos en la factoría de forma que los heliostatos obtendrán esa información cuando la necesiten.

3.2.3. Módulo de termopares

Comprueba una serie de temperaturas que son enviadas desde un sistema de control de temperaturas. Si alguna temperatura supera un cierto valor

de alarma, entonces el campo es desenfocado. Los valores de alarma de cada termopar son establecidos en el momento de arranque de la aplicación.

3.2.4. Módulos de disparo

Cada módulo de disparo se encarga de la comprobación de una tarjeta de entradas digitales. Cada módulo es una tarea RT-CORBA que lee periódicamente (cada 0,5 segundos) las entradas de la tarjeta y observa si existe alguna señal activa. Si es así desenfoca un porcentaje de los heliostatos que se encuentran enfocando al receptor. Este porcentaje viene establecido por una configuración que es cargada al iniciar la aplicación.

3.3. Subsistema de comunicaciones

El subsistemas de comunicaciones engloba todos aquellos módulos que precisan comunicación con el campo de heliostatos. Todos los módulos, incluyendo la factoría, son objetos RT-CORBA. En la figura 6 se muestra un esquema de éste.

3.3.1. Módulos de comunicaciones

Estos módulos, implementados como objetos pasivos, se encargan de gestionar las comunicaciones de todos los objetos heliostato conectados a una misma línea de comunicaciones. Puesto que un conjunto de objetos heliostato intentará comu-

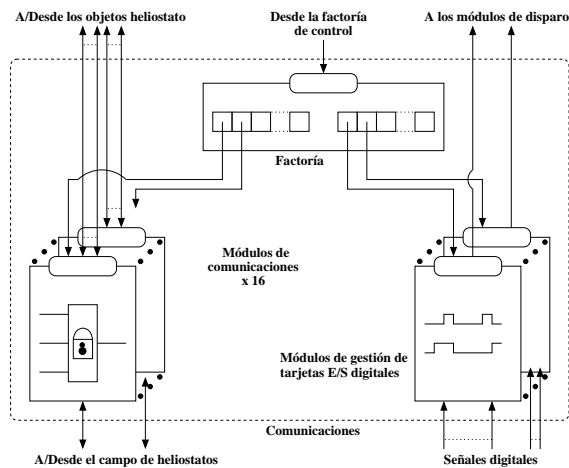


Figura 6: Esquema del subsistema de comunicaciones

nicar a través de un mismo canal de comunicación, posiblemente al mismo tiempo, se deberá secuenciar de forma correcta el acceso al canal para evitar errores en las comunicaciones debidas a condiciones de carrera.

Aunque depende de la tecnología usada, normalmente las comunicaciones suponen un mensaje desde el control central hacia el control local del heliostato físico e inmediatamente la respuesta de éste hacia el control central. Por tanto, mediante el uso de mutex se deben bloquear todas las peticiones de los objetos heliostato que intentan comunicar mientras un objeto heliostato esté comunicando con su equivalente físico. Una vez que el objeto heliostato que bloqueaba el módulo de comunicación finalice, liberará el mutex del módulo permitiendo a otro objeto heliostato acceder al canal de comunicación.

3.3.2. Módulos de gestión de tarjetas

Estos módulos gestionan el acceso a tarjetas de entradas digitales. La existencia de estos módulos permiten la distribución de la aplicación de forma que estos módulos se encuentran en el computador donde se encuentren las tarjetas, mientras que el resto de la aplicación puede encontrarse en otro computador.

3.4. Ajuste automático de offsets

Esta aplicación trata de calcular las desviaciones que se producen en los heliostatos durante la operación diaria de seguimiento de la trayectoria solar. Estas desviaciones son debidas a fallos en los motores, la naturaleza discreta de los codificadores ópticos incrementales, holguras en las mecanizaciones, movimientos producidos por fuertes vientos, etc.

Actualmente el ajuste de offsets se realiza de manera manual por parte del operador (el operador hace al heliostato apuntar a un punto predeterminado y corrige la desviación manualmente para que el rayo reflejado incida en el punto deseado). El objetivo de la aplicación ha sido desarrollar un sistema que permite llevar a cabo la corrección automática supervisada de los offsets de los heliostatos, automatizando esta tarea que hasta el momento se realizaba manualmente.

Este sistema se basa básicamente en la adquisición y procesado mediante técnicas de visión artificial de imágenes del reflejo del haz de luz solar en una diana situada en la torre. Si se trata de un heliostato desajustado el reflejo apuntará a un lugar de la torre fuera del punto deseado de la diana. Para la adquisición de las citadas imágenes se necesita una cámara situada en el centro del campo de heliostatos y enfocando la torre con la diana. Este programa necesita algún mecanismo para mover los heliostatos y corregir las desviaciones. Se ha preparado el programa para comunicarse tanto con el programa antiguo para el control del campo mediante un enlace RS-232 y para comunicarse usando CORBA con el nuevo programa para el control del campo.

El uso del nuevo servidor que usa CORBA ofrece muchas más posibilidades que el sistema actual (μ VAX) ya que permite tener realimentación. El nuevo servidor permite obtener información del heliostato mediante las llamadas CORBA, mientras que en el sistema actual sólo admite el envío de un subconjunto de comandos a ejecutar pero no hay ninguna forma de conocer el estado del heliostato. Con el nuevo servidor se puede, por ejemplo, conocer a que posición espacial está apuntando un heliostato, que azimut o elevación posee, que comando está ejecutando, etc. Además, se puede tener un mayor control sobre los heliostatos y realizar acciones sobre ellos como por ejemplo mandarlo a abatimiento o a adquisición permitiendo descender o elevar el reflejo del heliostato por un pasillo de seguridad.

3.5. Funcionamiento de la aplicación

El funcionamiento normal de la aplicación es el siguiente. Inicialmente se lanza los subsistemas de comunicaciones y de control. Estos funcionan como factorías para los objetos CORBA. Los objetos heliostato serán creados por medio de peticiones realizadas por parte de la interfaz y se les asignará un módulo de comunicaciones. Cuando se ejecute la interfaz, ésta leerá la información acerca del campo desde la base de datos e irá haciendo peticiones de creación de heliostatos al subsistema de control. Una vez iniciados todos los módulos y

creados los heliostatos, la interfaz irá obteniendo la información sobre el estado del campo y enviará los comandos al subsistema de control.

Una vez creada la infraestructura de objetos que forma la aplicación es posible considerar varias configuraciones según la potencia de cálculo disponible, el número de líneas de comunicaciones del campo, el número de heliostatos, etc.

A continuación se muestra una porción del código IDL que sirve de interfaz al subsistema de control.

```
module CCH{
/* Estado lógico del heliostato */
enum estadoLogico_CORBA_t{
    /* ABatido */
    CORBA_EST_AB,
    ...
    /* Seguimiento Normal */
    CORBA_EST_SN,
    /* Mando Manual */
    CORBA_EST_MM
};

/* Orientación del heliostato como
 * los valores del controlador de cada
 * motor */
struct orientHelio_CORBA_t {
    short elevac; /* Elevación */
    short azimut; /* Azimut */
};

interface heliostato {
    /* Envía a abatimiento pasando por
    ** un pasillo de seguridad */
    void abatir ();
    /* Envía a una orientación indicada
    ** en el parámetro */
    void mandoManual (in
        orientHelio_CORBA_t cons);
    ...
    /* Offsets del heliostato */
    attribute orientHelio_CORBA_t offsets;
    /* Orientación del heliostato */
    readonly attribute orientHelio_CORBA_t
        consignaActual;
};

interface control {
    /* Crea un heliostato */
    heliostato crearHeliostato (in
        paramHeliostato_CORBA_t param);
    ...
    /* Obtiene la referencia */
    heliostato obtenerHeliostato (in
        string id);
};
};
```

4. Conclusiones y trabajos futuros

Se ha descrito el diseño e implementación de un Sistema de Control para Campos de Heliostatos en donde se deben contemplar restricciones temporales, por tanto, para tenerlas en cuenta se está creando una aplicación de tiempo real distribuido. Aunque el SCCH no está completamente finalizado, partes importantes de éste ya se están testeando como por ejemplo los objetos CORBA que manipulan los heliostatos. Se han realizado ensayos satisfactorios en los que se controlan todos los heliostatos usando todas las línea de comunicaciones con la interfaz de usuario, estando la interfaz y los objetos heliostato en computadoras diferentes. A la vez que los objetos heliostato respondían a las peticiones de información y a los comandos emitidos desde la interfaz, también se atendían peticiones realizadas desde la aplicación para el ajuste automático de offsets.

Entre las partes importantes que quedan por finalizar en la aplicación están la manipulación de los eventos generados por el sistema (señales y termopares). Una vez terminada la implementación de la aplicación se pretenden realizar pruebas de rendimiento dependiendo del esquema de distribución utilizado y las distintas configuraciones que permite RT-CORBA, realizar análisis de planificabilidad para sistemas de tiempo real distribuidos y aplicar estrategias de control de apunte [3].

Inicialmente se tiene pensado ejecutar toda la aplicación (interfaz y subsistema de control) en un mismo computador para posteriormente ubicar la interfaz, el subsistema de control y el subsistema de comunicaciones en computadores diferentes. Si las necesidades de cómputo que precisa la planificabilidad de las tareas de tiempo real lo exigiera (aunque las pruebas realizadas hasta ahora no apuntan en ese sentido) se distribuirán los objetos heliostatos del subsistema de control en varios computadores, conectados éstos en una red Ethernet determinista local dedicada. La configuración inicial que se pretende usar para RT-CORBA incluye la transferencia de prioridades en las llamadas CORBA de manera que llamadas de ciertos clientes con mayor prioridad (como puede ser la tarea encargada de señalizar alarmas por temperaturas muy elevadas en el receptor) sean atendidas antes que otras. En el arranque de la aplicación se crearán todas las tareas para los heliostatos y se creará un *thread pool* para atender las peticiones de los clientes CORBA.

Referencias

- [1] J. D. Álvarez, M. Berenguel, A. Valverde, M. Romero, L. Yebra, E. F. Camacho, F. R. Ru-

- bio. *Diseño e implementación de los sistemas de control de un receptor solar volumétrico y de la etapa de potencia de una central eléctrica termosolar*. XXIV Jornadas de Automática, León. 2003.
- [2] M. Berenguel, F. R. Rubio, A. Valverde, P. J. Lara, M. R. Arahál, E. F. Camacho, M. López. *An artificial vision-based control system for automatic heliostat positioning offset correction in a central receiver solar power plant*. 2003. *Solar Energy*, 76, pp. 563-575.
- [3] F.J. García-Martín, M. Berenguel, A. Valverde, E. F. Camacho, *Heuristic Knowledge-based Heliostat Field Control for the Optimization of the Temperature Distribution in a Volumetric Receiver*. 1999. *Solar Energy*, 66(5), pp. 355-369
- [4] J. González. *Estudio e implementación de alternativas para el diseño y realización de sistemas de control de tiempo real con coste mínimo*. 2003. Proyecto Fin de Carrera. Universidad de Almería.
- [5] L. J. Yebra, M. Berenguel, M. Romero, D. Martínez, A. Valverde. *Automation of solar Plants*. EUROSUN 2004. The 5th ISES Europe Solar Conference, 20-23 June 2004, Freiburg, Germany.
- [6] D.C. Schmidt, S.D. Huston. *C++ Network Programming: Mastering Complexity Using ACE and Patterns*. 2003. Addison-Wesley Longman.
- [7] D.C. Schmidt, S.D. Huston. *C++ Network Programming: Systematic Reuse with ACE and Frameworks*. 2003. Addison-Wesley Longman.
- [8] D. Schmidt, D. Levine, C. Cleeland. *Architectures and Patterns for Developing High-performance, Real-time ORB Endsystems*. 1998
- [9] *TAO Developers Guide*. Object Computing, Inc., 2000.

AUTOMATIZACIÓN, CONTROL Y AUTOLOCALIZACIÓN DE UN VEHÍCULO QUAD

A. Valera, M. Mellado, I. Porris, E.J. Bernabeu

Dpto. Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia
giuprog@isa.upv.es, martin@isa.upv.es, igporgme@doctor.upv.es, ebernabe@isa.upv.es

Resumen

Los vehículos móviles conforman una plataforma muy utilizada en el área de la mecatrónica y el control de procesos. Algunos trabajos se centran en el desarrollo de nuevas plataformas móviles con, habitualmente, motores eléctricos. Otros se centran para proporcionar autonomía a vehículos existentes, como coches y camiones. Sin embargo, los quad son una plataforma experimental muy apropiada puesto que, además de tener las mismas características que los vehículos anteriores, tienen otras adicionales como menor peso y tamaño reducido.

La primera fase para la automatización del quad ha sido el diseño y el desarrollo de un sistema basado por computador para el control del acelerador, el freno y la dirección del quad. Para ello se ha tenido que diseñar el sistema mecánico, seleccionar los componentes e implementar el hardware y el software. Este artículo explica con detalle los distintos desarrollos llevados a cabo en esta fase.

Al mismo tiempo se ha equipado al sistema quad de un láser, utilizado como único sensor, para conseguir la autolocalización en entornos exteriores o interiores tipo garaje.

Palabras Clave: Robótica móvil, Control por computador, Conducción automatizada, Scan-matching, Autolocalización.

1 INTRODUCCIÓN

El campo de la conducción autónoma de vehículos es una aplicación investigación avanzada de robótica. Los vehículos de conducción autónoma mas modernos compiten una vez al año en el DARPA Grand Challenge [8]. El campeón del año 2005 tardó 6h 53min para completar las 132 millas de la carrera. El Stanford Vehicle [9] (llamado "Stanley") está basado en el vehículo de serie Volkswagen Touareg R5. Stanley está actuado vía sistema drive-by-wire desarrollado por Volkswagen en el "America's Electronic Research Lab". Todos los cálculos se realizan en varios computadores Pentium M

alimentados por un sistema de baterías controladas electrónicamente. El vehículo incorpora medidas desde varios GPS y unidades de medición inercial de 6 grados de libertad y de velocidad para la estimación de la posición.

Mientras que el vehículo está en movimiento se observa el entorno mediante 4 sistemas láser, un sistema radar, un par de cámaras estéreo y un sistema de visión monocular. Todos los sensores adquieren la información del entorno con una frecuencia entre 10 y 100Hz. La información del mapa y del posicionado se lee a 10Hz, permitiendo al vehículo evitar colisiones con obstáculos en tiempo real.

Los vehículos de 4 ruedas tienen un amplio rango de operaciones de campo abierto que no se pueden llevar a cabo con robots por su pobre movilidad en terrenos rugosos. Por ejemplo, para la exploración planetaria se necesitan robots con gran autonomía para la realización de misiones de grandes distancias.

Esta clase de vehículos son una de las plataformas mecatrónica más usadas [1], de forma que las misiones futuras cambiarán desde exploración pura a sistema de exploración real.

2 DESARROLLO DE LA AUTOMATIZACIÓN DE UN QUAD

La plataforma está basada en el vehículo quad 2003 mini ATV 49 de Macbor (Figura 1). El quad tiene un motor de 4 tiempos refrigerado por aire con un cilindro de 49.5ccm. El vehículo pesa 68.5kg. y tiene una capacidad de carga de 45kg. y alcanza una velocidad máxima de 25km/h.

La elección de este quad se basó en su bajo coste, la capacidad de conducción fuera de caminos y, sobre todo, por disponer de algunos sistemas de seguridad. Por ejemplo, tiene un sistema de parada de emergencia mediante control remoto.

En este apartado se va a presentar la automatización realizada en el vehículo móvil en los sistemas de aceleración, frenado y dirección.



Figura 1. Vehículo quad utilizado en el proyecto

2.1 AUTOMATIZACIÓN DEL SISTEMA DE ACCELERACIÓN.

Para controlar la velocidad del quad se ha controlado de forma automática el acelerador. El vehículo tiene una válvula para regular la entrada de gasolina en el motor. La válvula se controla mediante un cable conectado al manillar del quad. Cuando se estira del cable la válvula gira en un sentido horario. Cuando se deja de estirar del cable la válvula vuelve a su posición de reposo mediante un muelle conectado en el disco del acelerador, cortando de este modo la entrada de gasolina en el motor.

Se ha medido la fuerza necesaria en el acelerador, dando un valor de 3.5N.cm. Por ello un sevo pequeño es capaz de suministrar la fuerza necesaria. En este caso se ha optado por el servo de Hitec HS-300 que proporciona las siguientes especificaciones:

- Velocidad de operación: 0.14seg/60° a 6.0V
- Par de salida: 3.5kg•cm (6.0V)
- Peso y dimensiones: 44.5g, 41x30x36mm

La primera opción estudiada fue actuar directamente sobre el cable del acelerador, pero fue descartada por su complejidad mecánica. La opción final escogida ha eliminado el cable y fijar el servo directamente en el disco del acelerador, tal como se puede observar en el Figura 2.

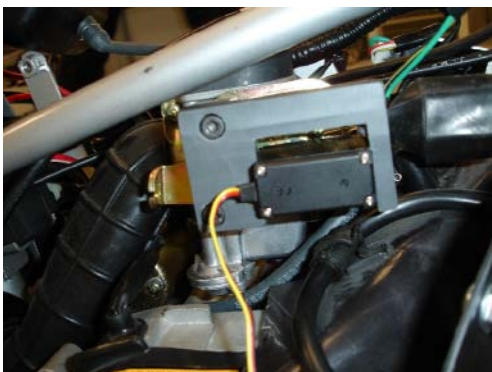


Figura 2. Ubicación del servo para la automatización del acelerador

2.2 AUTOMATIZACIÓN DEL SISTEMA DE FRENADO.

El quad viene equipado con un freno de tambor ubicado en la parte trasera. El freno se activa mediante un cable conectado en el manillar del quad. Apretando sobre dicho cable hace que las zapatas del freno se desplazan sobre el tambor, frenando de esta forma al vehículo

Para realizar la automatización del freno, se tuvo que medir primero la fuerza necesaria, dando un valor de, al menos, 32N.cm. Además, también fue necesario escoger la ubicación y montar el sistema de control.

La primera opción estudiada se basaba en un actuador lineal que apretaba al freno. Sin embargo dicha solución fue descartada debido a problemas de implementación y a su alto coste.

La solución final escogida se basa en un servo conectado al cable del freno. Esto no sólo requería modificaciones menores en el quad sino que además permite que el sistema original de freno siga funcionando. El servo escogido para realizar esta automatización es el HS-805BB de Hitec, que proporcionaba las siguientes especificaciones:

- Velocidad de operación: 0.14seg/60° a 6.0V
- Par de salida: 24.7kg•cm (6.0V)
- Peso y dimensiones: 152g, 60x30x57.6mm

Además del servo, para montarlo, se han tenido que fabricar algunas piezas. La Figura 3 muestra con mayor detalle como se ha resuelto la automatización del sistema de freno.



Figura 3. Sistema de frenado del quad desarrollado

2.3 AUTOMATIZACIÓN DEL SISTEMA DE DIRECCIÓN.

Una de las premisas de que se partía para el diseño de la automatización del sistema de dirección fue la de requerir sólo cambios menores.

Se ha medido la fuerza de rotación en el manillar, dando un momento entre 15N.m (para superficies

suaves) y 20 N.m (para superficies rugosas). Para evitar posibles problemas se ha asumido para los cálculos un valor de 30N.m

La automatización del sistema de dirección fue mucho más compleja que los casos de la aceleración y del freno. Por ello se estudiaron diferentes soluciones basadas en la combinación de motores eléctricos y cajas de engranajes conectados al eje de dirección. Sin embargo todas estas opciones fueron descartadas debido a su complejidad mecánica.

Finalmente, la opción escogida está basada en un actuador lineal puesto que no sólo era la solución más económica motor-caja de engranajes que podría proporcionar la fuerza necesaria, sino que además era la solución más simple para fijar el actuador lineal al quad.

Para desarrollar la automatización de la dirección sin necesitar cambios significativos en el quad se han tenido que diseñar y fabricar diferentes piezas y soportes, tal como se puede intuir en la Figura 4.

De acuerdo con la fuerza necesaria y la limitación del espacio (motor montado verticalmente a la unidad lineal), se escogió el actuador lineal ALI1-F MO3 de Tecnopower, proporcionando las características siguientes:

- Fmax: 1200N
- Velocidad lineal: 12 mm/s
- Velocidad del motor: 6000 rpm
- Factor de reducción de la caja de engranajes: 1:18

El extremo del eje del actuador lineal está fijo a la suspensión de la rueda derecha, de manera que se controla el giro de las ruedas controlando la posición del eje del actuador.

Para medir la posición del sistema de dirección se ha fijado un potenciómetro (Spectrol 357-0-0-102) a la base del eje del manillar.

2.4 COMENTARIOS FINALES.

Esta sección ha mostrado la automatización realizada para el sistema de aceleración, freno y dirección. La restricción de partida para esto fue no necesitar cambios significativos en la estructura del quad. Para ello, para cada tarea se diseñaron diferentes soluciones, escogiéndose finalmente aquella que minimizaba los cambios estructurales en el quad. De esta forma se ha logrado conservar la configuración de serie del vehículo. Así, todos los componentes añadidos en el vehículo se han fijado a la estructura del quad. La única modificación significativa ha sido eliminar el cable del acelerador.

Se puede considerar que la automatización del quad no ha sido una tarea muy complicada, sin embargo no ha sido simple ni trivial encontrar la solución mecánicamente fácil de montar, económica y robusta.

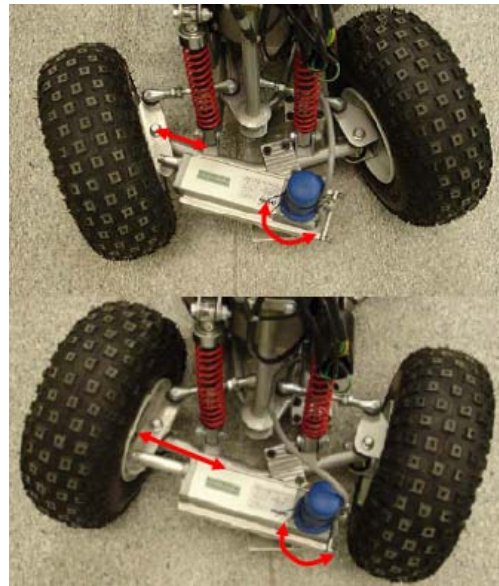


Figura 4. Sistema de control de la dirección basado en un actuador lineal

3 ARQUITECTURA DE CONTROL

Para poder llevar a cabo el control por computador del quad se ha tenido que implementar una arquitectura hardware/software de control. Este apartado presentará brevemente la arquitectura propuesta para la plataforma desarrollada.

3.1 ARQUITECTURA HARDWARE.

Para poder disponer de un sistema completamente abierto y flexible, la arquitectura hardware de control del vehículo móvil está basada en un computador personal. Además, para tratar de reducir el peso, se ha optado por un computador portátil equipado con un procesador Pentium M. La Figura 5 muestra con detalle las distintas etapas y componentes que conforman la arquitectura hardware de control.

El computador está equipado por un dispositivo de adquisición de datos USB de Nacional Instruments: el NI USB-6009. Se trata de un dispositivo multifunción con 8 canales de entrada analógica de 14 bits, 2 salidas analógicas, 12 canales de entrada/salida digital y un contador.

Este dispositivo se alimenta directamente del bus USB, por lo tanto no necesita ningún suministro de

tensión externa para trabajar. Además, incluye también 2 terminales desmontables para conectar directamente las distintas señales del sistema. Cabe destacar que el dispositivo de adquisición de datos también proporciona una tensión de referencia de salida que permite, por ejemplo, alimentar otros dispositivos y sensores.

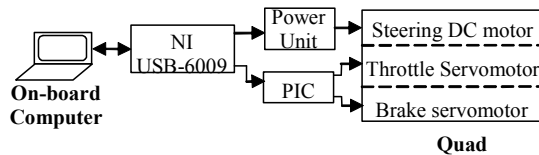


Figura 5. Arquitectura hardware de control

Debido a que el actuador lineal del sistema de dirección consume corrientes de más de 5A [10], la arquitectura de control necesita una etapa de potencia. Ésta está basada en la etapa de servoamplificador para motores de corriente continua ELMO SSA-12/55. La unidad SSA es un servoamplificador de PWM miniatura para motores de escobillas de hasta 850W. El amplificador utiliza transistores MOSFET y componentes SMT que proporcionan una corriente máxima continua de 12A, siendo la corriente máxima de pico de 30ª [2]. La frecuencia de conmutación del PWM es estándar de 20KHz.

Para establecer el control de la dirección del quad, la unidad de potencia necesita una tensión de referencia para especificar la velocidad y el sentido de giro de la dirección. Esta tensión de control la suministra el computador personal mediante un canal del convertidor Digital/Analógico de la tarjeta USB.

Por ultimo, para poder saber el giro de la dirección se ha equipado el quad con un potenciómetro alojado en la parte inferior de la dirección. El potenciómetro se alimenta directamente del dispositivo USB, y se lee la tensión (asociada al giro de las ruedas) con uno de los canales del convertidor Analógico/Digital.

La Figura 6 muestra con más detalle la parte de la arquitectura de control del sistema de dirección del quad.

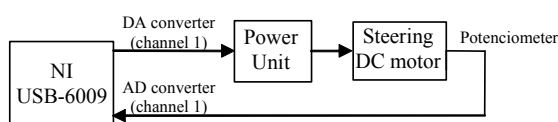


Figura 6. Arquitectura de control de la dirección

Para el acelerador y el freno, la arquitectura de control utiliza un microcontrolador. A partir de la tensión de control suministrada por el computador a partir de la unidad de adquisición de datos (utilizando

el 2º canal de salida analógica que tiene el dispositivo USB), el microcontrolador genera 2 señales PWM para los servomotores instalados en el quad. El microcontrolador utilizado es el PIC 16F876, de Microchip. Éste pertenece a la familia mediana de dispositivos PICmicro, y está equipado con 3puertos de entrada/salida, 2 módulos PWM, comunicaciones serie y 5 canales de entrada Analógico/Digital.

El primer servomotor controla el freno accionando el cable. El servo tiene internamente un potenciómetro, por lo tanto únicamente necesita la señal PWM generada por el microcontrolador para poder controlar la frenada del vehículo.

El servomotor del acelerador se controla mediante el 2º modulo PWM del microcontrolador. Como en el caso del freno, el PIC utiliza la tensión de control proporcionada por el 2º canal de salida analógica del dispositivo USB para generar la señal de PWM.

Además de la tensión analógica, el PIC necesita una señal más. Se trata de una señal digital proporcionada también por el dispositivo USB y que especifica si se trata de una tensión para controlar el freno o el acelerador. La Figura 7 muestra la arquitectura de control para las etapas del freno y acelerador.

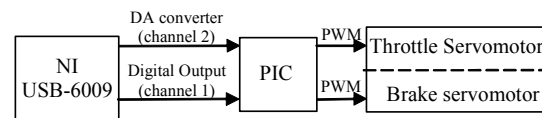


Figura 7. Arquitectura de control del freno y acelerador

3.2 ARQUITECTURA SOFTWARE.

La arquitectura presentada en la sección anterior tiene 2 ventajas principales: proporciona una arquitectura de control completamente abierta y flexible. Además, no representa un coste muy elevado. Utilizando esta arquitectura de control se puede implementar cualquier estrategia de control lo que permite al quad trabajar en tareas de alto nivel como, por ejemplo, la generación automática de trayectorias, planificación de movimientos, navegación por visión artificial, etc.

Para la programación del sistema de control se emplea el entorno Visual C++. Para poder acceder al dispositivo de adquisición de datos se utiliza el software NI-DAQmx Base. Éste es un driver multiplataforma de National Instruments para el desarrollo de aplicaciones de adquisición de datos con NI LabVIEW o entornos de desarrollo basados en lenguaje C.

Además del compilador de C, para el desarrollo de algunas aplicaciones se ha utilizado el paquete LabVIEW. Con este software se ha implementado una aplicación que permite conducir el quad

mediante un Pocket PC. Éste se comunica con el computador de control mediante tecnología wi-fi, enviando las referencias para la dirección, freno y aceleración. Para ello se ha utilizado el módulo LabVIEW PDA, que extiende el desarrollo de aplicaciones a dispositivos de mano. Con este módulo primero se implementa la aplicación en el entorno LabVIEW, descargándola posteriormente en cualquier dispositivo PDA de una forma rápida y directa [11].

El sistema de control basado en PC permite, además, la conexión de las estructuras de datos con paquetes comerciales CADCS como MATLAB, MATHEMATICA, etc.

De hecho, si se utilizan las últimas versiones de MATLAB, se puede acceder directamente al dispositivo USB de adquisición de datos cargando e invocando librerías dinámicas mediante comandos como loadlibrary, libpointer, calllib, etc.

Por último, para programar el microcontrolador que maneja los servomotores del freno y acelerador, se necesita un compilador de C y ensamblador, así como de un entorno para transferir el programa compilado desde el computador de desarrollo al microcontrolador.

4 AUTOLOCALIZACIÓN

Para planificar el movimiento del quad por entornos exteriores o interiores, de características similares a garajes, se necesita autolocalizar el vehículo. Uno de los métodos más satisfactorios para resolver este problema en entornos interiores es el scan-matching. No obstante, el problema se puede presentar en entornos exteriores especialmente con falta de obstáculos detectables por el sistema láser.

El sistema de scan-matching permite a la vez obtener mapas del entorno por el que circula el vehículo [4]. A partir de este mapa se facilitan capacidades para la autonomía del vehículo como son la planificación de caminos y otras de más alto nivel.

4.1 SCAN-MATCHING

El scan-matching es una técnica para mejorar la odometría conseguida por los encoders situados en las ruedas. Consiste en encajar dos tomas de datos del mismo entorno en posiciones diferentes, estas deben presentar puntos o zonas en común. Normalmente se efectúa sobre la distancia entre el vehículo y los objetos del entorno, leídos por un sensor de tal propósito basado en láser o sonar. Si se es capaz de encontrar la traslación y rotación entre una toma, que denominaremos toma de referencia, y

otra que llamaremos, toma nueva, entre los puntos o las zonas que tienen en común dichas tomas, se habrá hallado el vector distancia que se ha desplazado el vehículo entre las dos tomas y el giro realizado. El método de referencia utilizado por la comunidad, para resolver el problema del scan-matching, es el ICP (Iterative Closest Point) propuesto por Lu y Milios [4]. Se trata de un método iterativo que minimiza la energía de una función, que depende de la distancia entre los puntos de la toma de referencia y la nueva.

Para la aplicación realizada aquí, autolocalización de un vehículo con motor de gasolina, se ha aplicado un método mejorado llamado MbICP (Metric based iterative closest point) [6] que es más estable a rotaciones que el ICP. Este método se basa en el ICP, pero utiliza una definición de métrica que tiene en cuenta la rotación, por lo que es más estable a grandes rotaciones, uno de los principales problemas del ICP.

4.2 ESCENARIO APLICACIÓN

En primer lugar, las pruebas para comprobar la validez del método de scan-matching se realizan sobre un programa de simulación de robótica móvil, Webots [7]. Para ello, se crea un entorno en el que se ubica un robot diferencial con un sensor de distancia simulando un dispositivo láser como el que se usará posteriormente en el quad. Adicionalmente, en dicho entorno virtual, se crean paredes y pilares que podrían encontrarse en un parking de vehículos típico. Posteriormente, se simula la superficie externa de un coche estacionado a la izquierda del robot. Finalmente, se sitúa el robot y se ejecuta la simulación (Figura 8).

El programa de simulación Webots genera la dinámica en cada momento que actúa sobre el robot. Para el movimiento del robot se ha programado un controlador en C. Dicho controlador es ejecutado con una frecuencia de muestreo de 32 ms. El dispositivo láser construido tiene un alcance limitado, siendo lineal con la distancia. Su alcance es configurable, devolviendo el valor máximo cuando no se detecta nada. La apertura del láser es de 100°, tomándose una medida cada incremento de un ángulo.

Los datos que proporcionan el sensor se representan en el espacio virtual por segmentos rojos, cuando se encuentra espacio libre, y verdes, después de haber tocado algún objeto (Figuras 8 y 9).

Sobre este montaje se realizan varias experimentaciones para autolocalizar el robot, que se muestra en el siguiente apartado.

Una vez comprobado satisfactoriamente los resultados obtenidos en la simulación, se procede a hacer una prueba real de autocalización con el quad en un entorno de parking de automóviles. Para ello, se monta un sensor láser de rango Sick LMS 400 que presenta un alcance de 3 metros con una apertura de 75° y una resolución de 0.25° . Este láser se caracteriza por devolver el mínimo valor cuando no detecta nada.

En la experimentación real, se toman datos de un entorno típico de un parking, con pilares, paredes de hormigón y coches. Los resultados de la correspondiente autocalización se muestran en siguiente apartado.

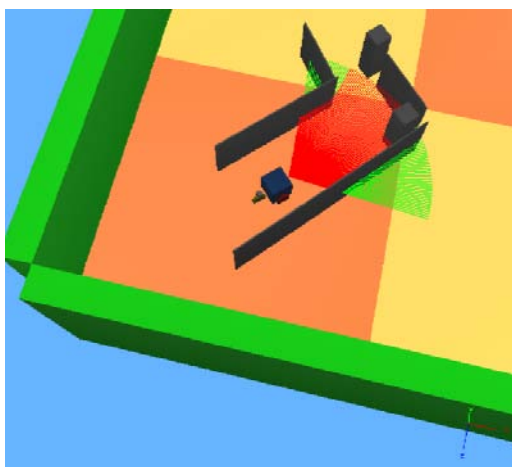


Figura 8.- Entorno generado para las pruebas de scan-matching, en el Webots.

4.3 RESULTADOS

En este apartado se muestran los resultados obtenidos durante el proceso de autocalización, tanto para la simulación realizada sobre Webots, como para el caso real. Para ello se ha utilizado la implementación del método MbICP disponible en [3].

En la experimentación realizada sobre simulador, se muestran tres instantáneas del movimiento del robot en la Figura 9. Durante ese movimiento, el sistema láser ha ido proporcionando información del entorno que ha sido procesada por el algoritmo MbICP.

En la Figura 10, se muestra las posiciones por la que ha ido pasando el robot y la zona libre dando pie a la generación de un mapa. Así pues, los puntos amarillos indican puntos sin explorar, los puntos verdes espacio libre y los puntos rojos que se ha detectado un reflejo en ese punto. Los segmentos azules representan la posición y orientación, calculado por el algoritmo, de cada uno de los scans.

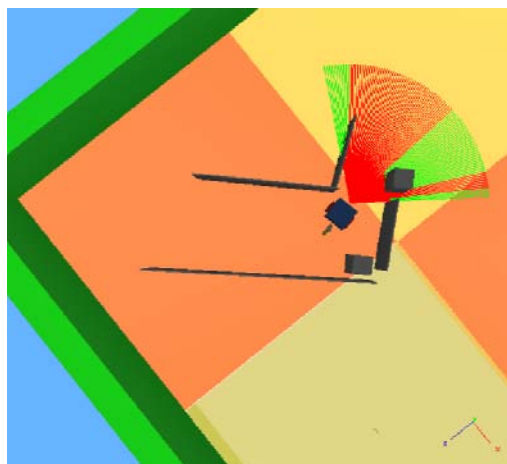
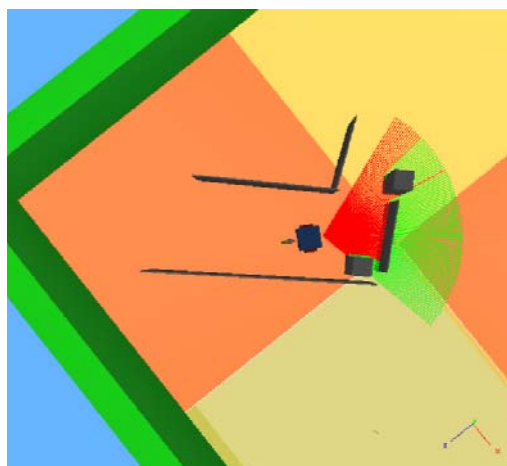
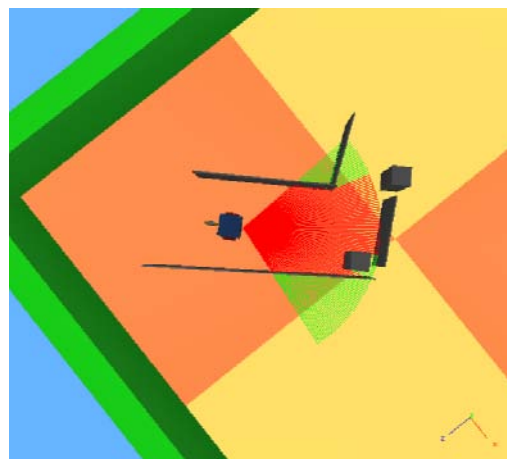


Figura 9.- Progreso del robot en el entorno virtual, primera imagen posición inicial, última imagen posición final.

Para la aplicación real sobre el parking, de los datos proporcionados por el láser, se ha obtenido el resultado que se muestra en la figura 11. Donde cada punto equivale en la realidad a 1 cm^2 , con lo que, si se tienen en cuenta las dimensiones del quad, la precisión es muy alta. Como consecuencia, se puede determinar un mapa de rejilla, que permitirá en un

futuro agregar nuevas capacidades al quad para conseguir su total autonomía.

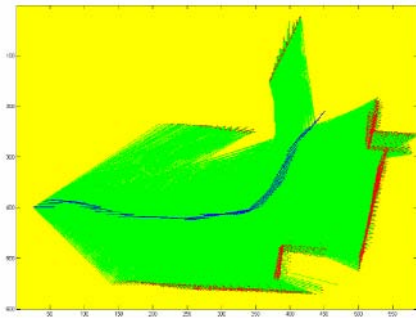


Figura 10.- Mapa generado con los datos generados por el robot y entorno virtual.

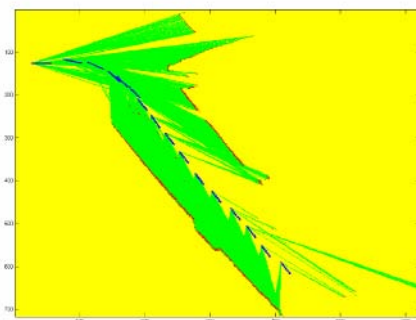


Figura 11.- Mapa generado con los datos adquiridos en un entorno real.

5 CONCLUSIONES Y TRABAJOS FUTUROS

El artículo ha presentado una experiencia práctica para la automatización de un vehículo quad. Para establecer el control automático del quad se han motorizado el acelerador, el freno y la dirección del vehículo de acuerdo a la arquitectura hardware y software explicada en el artículo.

Al mismo tiempo, se le ha equipado de un sistema que permite obtener los primeros resultados de autolocalización para un vehículo quad, que puede desplazarse en entornos exteriores o interiores de tipo parking. Para resolver este problema sólo se ha usado un sensor de tipo láser. El método para autolocalizar usado ha sido el Metric-based Iterative Closest Point.

Antes de aplicar la autolocalización en un parking real, se diseñó un entorno similar mediante el uso del software comercial Webots. Tanto en la aplicación simulada como en la real, se han obtenido resultados satisfactorios. En definitiva, se puede concluir que este tipo de dispositivo láser es adecuado para el entorno real considerado.

Este trabajo, se encuentra embebido dentro de un proyecto de investigación, financiado por el gobierno autónomo valenciano. Su objetivo es la conducción totalmente autónoma del vehículo quad. Fases siguientes de la investigación iniciada van a ser la detección en exteriores de obstáculos, tanto estáticos como dinámicos. Un aspecto importante va a ser estimar la velocidad de estos últimos, y así planificar la velocidad y dirección del movimiento del quad para evitar la colisión con todo tipo de obstáculo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos FEDER-CICYT con referencia DPI2005-08732-C02-02 y DPI2005-09133-C03-01, del Ministerio de Educación y Ciencia, así como por el proyecto de investigación de la Generalitat Valenciana, Consellería de Empresa, Universidad y Ciencia GV06/115.

Los autores quieren también agradecer a los estudiantes que han participado en el desarrollo de parte de este proyecto, especialmente a los estudiantes Erasmus A. Ehrmann y S. Sandhass, de la Technical University of Kaiserslautern. Sin olvidar, la inestimable colaboración de Javier Mínguez, cediendo la utilización de la implementación de su técnica de scan-matching.

Referencias

- [1] W. Bolton, "Mechatronics. Electronic Control Systems in Mechanical and Electrical Engineering", Person Education Limited, ISBN 0-582-35705-5.
- [2] Elmo Motion Control. SSA Series. Installation & Operating Manual.
- [3] P. Jensfelt, J. Mínguez, "Implementación de Algoritmo MbICP", versión software de 28-03-2006
- [4] F. Lu, E. Milios, "Globally consistent range scan alignment for environment mapping," Autonomous Robots, vol. 4, pp. 333-349, 1997.
- [5] Microchip. PIC16F87/88 Data Sheet.
- [6] J. Mínguez, F. Lamiroux, L. Montesano, "Metric-based Scan Matching Algorithms for Mobile Robot Displacement Estimation", IEEE Int. Conf. on Robotics and Automation, pp.3568-3574, Barcelona, 2005
- [7] Webots (<http://www.cyberbotics.com>), un software de simulación para robots móviles desarrollado por Cyberbotics Ltd.
- [8] <http://www.darpa.mil/grandchallenge/index.asp>
- [9] <http://www.stanfordracing.org/>
- [10] <http://www.tecnopower.es>
- [11] <http://www.ni.com/>

MODELADO E IDENTIFICACIÓN DE PARÁMETROS DINÁMICOS DE ROBOTS. APLICACIÓN A UN ROBOT PUMA 560

N. Rosillo, F. Benimeli, A. Valera[‡], V. Mata[†]

nurogue@mcm.upv.es, frabeau@iti.upv.es, giuprog@isa.upv.es, vmata@mcm.upv.es

[‡]Dpto. de Ingeniería de Sistemas y Automática, [†]Dpto. de Ingeniería Mecánica y de Materiales
Universidad Politécnica de Valencia

Resumen

Este trabajo presenta el desarrollo y validación de una metodología que permita la identificación de parámetros dinámicos en robots manipuladores. Las ecuaciones dinámicas de los robots están basadas en las ecuaciones de movimiento de Gibbs-Appell, proporcionando unas ecuaciones bien estructuradas que, además de poder ser calculadas en tiempo real, permiten expandir fácilmente el modelado, lo que permite incorporar el efecto de los actuadores o el fenómeno de la fricción.

El artículo aborda posteriormente la identificación del modelado dinámico de robots manipuladores, proponiendo 2 métodos distintos, validándose sobre un sistema robotizado industrial: un robot PUMA 560 dotado de una arquitectura abierta de control basado en un computador personal industrial.

Palabras Clave: Modelado de robots, identificación de robots, control de robots, robots manipuladores.

1 INTRODUCCIÓN

La identificación de los parámetros dinámicos que constituyen el modelo dinámico de un robot manipulador tiene por objeto la estimación de valores precisos de dichos parámetros a partir de medidas experimentales del movimiento del robot, siendo éste el único procedimiento práctico que permite la obtención de valores fiables de los mismos cuando el sistema mecánico presenta una mínima complejidad. La importancia de la identificación de parámetros dinámicos se manifiesta especialmente tanto en aplicaciones de control por dinámica inversa como en simulación dinámica. En este trabajo se aborda la identificación de parámetros dinámicos en robots manipuladores con configuración de cadena cinemática abierta.

Por una parte, se desarrolla el modelo dinámico de un robot manipulador a partir de las ecuaciones de la dinámica de acuerdo al formalismo de Gibbs-Appell. Para ello, se asume el robot constituido por elementos rígidos, modelándose independientemente el comportamiento dinámico de los actuadores. Se consideran asimismo algunos modelos de fricción

lineales con respecto a sus coeficientes con objeto de modelar los fenómenos de fricción en los nudos. Posteriormente, las ecuaciones que constituyen dicho modelo dinámico son rescritas en forma lineal respecto a los parámetros dinámicos a identificar y expresadas en forma matricial, a fin de permitir la posterior aplicación de técnicas numéricas tanto de análisis y reducción como de resolución del sistema de ecuaciones así constituido.

Para poder obtener una correcta identificación de parámetros dinámicos es necesario disponer de un conjunto de trayectorias con poder de excitación de los algoritmos de identificación. En este trabajo se proponen la generación de trayectorias optimizadas, recurriéndose a la parametrización de las mismas mediante series finitas de Fourier, lo cual permite beneficiarse del carácter periódico de éstas.

De este modo se plantean dos métodos de identificación: el primero de ellos de acuerdo al esquema de identificación directo, siendo el segundo una combinación entre los esquemas de identificación directo e indirecto.

Tanto el modelo dinámico del robot manipulador como los distintos métodos de identificación propuestos se validan mediante su aplicación experimental a la identificación de los parámetros dinámicos de un robot manipulador industrial PUMA 560. Así mismo, las técnicas propuestas son validadas también mediante el desarrollo y la ejecución de varias estrategias de control, estrategias basadas en las ecuaciones dinámicas cuyos términos se identifican. Para ello se ha dotado al robot industrial de una arquitectura de control abierta basada en un computador personal industrial que permite, mediante lenguaje C, el desarrollo de aplicaciones de control en tiempo real.

2 MODELADO DINÁMICO DE ROBOTS MANIPULADORES

Para generar las ecuaciones del movimiento de un sistema mecánico (un robot manipulador en este caso) hace falta obtener su modelo dinámico. Para ello, como es bien sabido, se pueden utilizar diferentes principios de la dinámica, como por ejemplo Newton-Euler, Jourdain o Lagrange-Euler.

En este trabajo se han utilizado las ecuaciones de Gibbs-Appell en virtud de que, pese a su relativamente escasa utilización en el campo de la Robótica, constituyen una de las formas más simples de las ecuaciones de la Dinámica. Esta formulación permite expresar las ecuaciones del robot en forma lineal respecto a los parámetros dinámicos, lo que es muy interesante desde el punto de vista de la identificación de dichos parámetros. Las ecuaciones de Gibbs-Appell aportan otras ventajas adicionales, como por ejemplo facilitar la inclusión del efecto de los actuadores [6].

Las ecuaciones de Gibbs-Appell se derivan de la definición de la función de Gibbs, también conocida como la energía de las aceleraciones. Cuando se escribe en su forma original para un sólido cualquiera compuesto de n partículas elementales con masa m_i y aceleración a_i y considerando un sistema de referencia inercial, ésta es:

$$G = \frac{1}{2} \sum_{i=1}^n m_i a_i^2 \quad (1)$$

En el caso de un robot manipulador compuesto por n barras consideradas como sólidos rígidos, la función de Gibbs para la barra i viene dada por:

$$G_i = \frac{1}{2} \cdot m_i \cdot {}^0\ddot{r}_{G_i}^T \cdot {}^0\ddot{r}_{G_i} + \frac{1}{2} \cdot {}^0\dot{\omega}_i^T \cdot {}^0\mathbf{I}_{G_i} \cdot {}^0\dot{\omega}_i + {}^0\dot{\omega}_i^T \cdot [{}^0\bar{\omega}_i \times ({}^0\mathbf{I}_{G_i} \cdot {}^0\bar{\omega}_i)] \quad (2)$$

donde m_i es la masa de la barra, ${}^0\mathbf{I}_{G_i}$ es su tensor centroidal de inercia, ${}^0\bar{\omega}_i$ y ${}^0\dot{\omega}_i$ representan respectivamente la velocidad y aceleración angulares de la barra y ${}^0\ddot{r}_{G_i}$ es la aceleración del centro de masas de la barra. Todas estas magnitudes están expresadas considerando un sistema de referencia inercial.

Las ecuaciones del movimiento de Gibbs-Appell se obtienen derivando la función de Gibbs expresada anteriormente respecto a las aceleraciones generalizadas \ddot{q}_k , obteniendo de esta manera las fuerzas de inercia generalizadas que compensan las fuerzas exteriores generalizadas, esto es τ_k :

$$\tau_k = \sum_{i=k}^n \frac{\partial G_i}{\partial \ddot{q}_k}, \quad k = 1, 2 \dots n \quad (3)$$

Desarrollando la expresión, quedará:

$$\tau_k = \sum_{i=k}^n \left\{ m_i \cdot {}^i\ddot{r}_{G_i}^T \cdot \frac{\partial {}^i\ddot{r}_{G_i}}{\partial \ddot{q}_k} + {}^i\dot{\omega}_i^T \cdot {}^i\mathbf{I}_{G_i} \cdot \frac{\partial {}^i\dot{\omega}_i}{\partial \ddot{q}_k} + \left(\frac{\partial {}^i\dot{\omega}_i}{\partial \ddot{q}_k} \right)^T \cdot [{}^i\bar{\omega}_i \times ({}^i\mathbf{I}_{G_i} \cdot {}^i\bar{\omega}_i)] \right\} \quad (4)$$

Reorganizando los elementos de la expresión (4), pueden identificarse dos términos distintos, tal como se aprecia en la siguiente expresión:

$$\tau_k = \underbrace{\sum_{i=k}^n \left\{ \left(\frac{\partial {}^i\dot{\omega}_i}{\partial \ddot{q}_k} \right)^T \cdot [{}^i\mathbf{I}_{G_i} \cdot {}^i\dot{\omega}_i + {}^i\bar{\omega}_i \times ({}^i\mathbf{I}_{G_i} \cdot {}^i\bar{\omega}_i)] \right\}}_{A_k} + \underbrace{\sum_{i=k}^n \left\{ \left(\frac{\partial {}^i\ddot{r}_{G_i}}{\partial \ddot{q}_k} \right)^T \cdot m_i \cdot {}^i\ddot{r}_{G_i} \right\}}_{B_k} \quad (5)$$

Para poder abordar el proceso de identificación de los parámetros inerciales de las ecuaciones dinámicas de los robots se necesita la reestructuración de estas ecuaciones de manera que los parámetros a identificar aparezcan de forma lineal en las mismas.

Partiendo de los términos A_k y B_k , sustituyendo la expresión de la aceleración lineal del centro de gravedad de la barra i -ésima, tomando en consideración el teorema de los ejes paralelos con el fin de expresar los tensores de inercia respecto al origen de los sistemas de referencia locales se llega a la siguiente expresión de las fuerzas generalizadas linealizadas respecto a los parámetros inerciales:

$$\tau_{l_k} = \left(\frac{\partial {}^k\dot{\omega}_k}{\partial \ddot{q}_k} \right)^T \sum_{i=k}^n {}^k\mathbf{R}_l [{}^i\hat{\eta}_i \cdot {}^i\bar{l}_i + ({}^i\tilde{r}_{O_k O_i} {}^i\tilde{\eta}_i - {}^i\tilde{r}_{O_i} \cdot {}^i\tilde{r}_{O_i}) \cdot m_i {}^i\tilde{r}_{O_k G_i} + {}^i\tilde{r}_{O_k O_i} {}^i\ddot{r}_{O_i} \cdot m_i] \quad (6)$$

donde, por ejemplo:

$${}^k\hat{\eta}_i = ({}^k\dot{\omega}_i + {}^k\tilde{\omega}_i {}^k\hat{\omega}_i), \quad {}^k\tilde{\eta}_i = ({}^k\dot{\omega}_i + {}^k\tilde{\omega}_i {}^k\tilde{\omega}_i)$$

Por consiguiente, el sistema de ecuaciones correspondiente al modelo de barras presentará, expresado en su forma matricial, la siguiente forma para una configuración $(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}})$ dada del robot manipulador:

$$\bar{\tau}_l = \mathbf{W}_l(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi}_l \quad (7)$$

donde $\bar{\tau}_l$ es el vector de fuerzas generalizadas, \mathbf{W}_l es la matriz de observación y $\bar{\Phi}_l$ es el vector de parámetros inerciales del sistema, el cual presenta, en este caso, la siguiente forma:

$$\bar{\Phi}_l = [{}^n\bar{l}_n \quad \dots \quad {}^1\bar{l}_1 \quad m_n \cdot {}^n\tilde{r}_{O_n G_n} \quad m_n \quad \dots \quad m_1 \cdot {}^1\tilde{r}_{O_1 G_1} \quad m_1]^T$$

Para incrementar el grado de precisión en la determinación de los parámetros del robot manipulador se puede incorporar el modelo dinámico

de los rotores. Este modelo dinámico tiene por objeto la modelización del par correspondiente al efecto giroscópico de los rotores, así como de eventuales excentricidades del eje de giro de los mismos.

Procediendo análogamente al caso del modelo de los elementos rígidos, la expresión del par debido a los rotores calculado a partir de la definición de la función de Gibbs para el caso de los rotores vendrá dada por:

$$\tau_k = \sum_{i=1}^n \left[\left(\frac{\partial^i \dot{\omega}_{r_i}}{\partial \ddot{q}_k} \right)^T \left({}^{r_i} \mathbf{I}_{G_{r_i}} \cdot {}^{r_i} \dot{\omega}_{r_i} + {}^{r_i} \dot{\omega}_{r_i} \times {}^{r_i} \mathbf{I}_{G_{r_i}} \cdot {}^{r_i} \dot{\omega}_{r_i} \right) + \left(\frac{\partial^i \ddot{r}_{G_{r_i}}}{\partial \ddot{q}_k} \right)^T m_{r_i} \cdot {}^{r_i} \ddot{r}_{G_{r_i}} \right] \quad (8)$$

Teniendo en cuenta las expresiones de las variables cinemáticas de los rotores, utilizando el teorema de los ejes paralelos y asumiendo que el actuador r_i que actúa sobre la barra k está situado en una de las primeras k barras (tal como es habitualmente el caso en la mayoría de robots manipuladores industriales), se puede obtener el sistema de ecuaciones correspondiente al modelo dinámico de los rotores, expresado en su forma matricial, el cual presentará la siguiente forma:

$$\bar{\tau}_r = \mathbf{W}_r(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi}_r \quad (9)$$

donde $\bar{\tau}_r$ es el vector de fuerzas generalizadas, \mathbf{W}_r es la matriz de observación y $\bar{\Phi}_r$ es el vector de parámetros dinámicos del sistema, el cual presenta, en este caso, la siguiente forma:

$$\bar{\Phi}_r = \begin{bmatrix} {}^{r_1} \bar{J}_{r_1}^T & \dots & {}^{r_{n_1}} \bar{J}_{r_{n_1}}^T & m_{r_n} \cdot {}^{r_n} \bar{r}_{O_n G_{r_n}}^T & m_{r_n} & \dots & m_{r_1} \cdot {}^{r_1} \bar{r}_{O_1 G_{r_1}}^T & m_{r_1} \end{bmatrix}^T$$

Además de modelar los elementos y los actuadores de los robots, también se ha obtenido la modelización de la fricción. El fenómeno de la fricción puede aparecer debido a una gran variedad de mecanismos físicos, como puedan ser la deformación elástica o plástica en el contacto entre superficies de partes mecánicas implicadas en el movimiento, y se ve además muy influenciado por factores internos como la presencia de lubricantes o sustancias contaminantes y su grado de distribución, así como indirectamente por otros factores como las variaciones térmicas, principalmente debidas al propio funcionamiento del robot.

No resulta posible deducir un modelo de fricción simple a partir de principios físicos elementales. En su lugar, suele recurrirse a describir la fricción a partir de una serie de fenómenos observables desde un punto de vista macroscópico. En este sentido, los

modelos de fricción se dividen en dinámicos y estáticos, según que el efecto de la variación de velocidad sea considerado o no. Dentro de los modelos estáticos, se encuentra una serie de modelos clásicos en los cuales se asume una mayor o menor idealización del fenómeno, donde se consideran dos componentes de la fricción: por un lado, un término constante (el bien conocido término de Coulomb) que no depende, por tanto, de la velocidad y, por otro lado, la llamada fricción viscosa, cuya variación será lineal con la velocidad.

En el presente trabajo se ha optado por utilizar modelos de fricción cuyos coeficientes aparezcan de forma lineal en el modelo ya que así se posibilita la integración del modelo de fricción en el modelo dinámico global del manipulador en su forma matricial, lo cual resulta enormemente útil a la hora de llevar a cabo una identificación directa. Se han considerado cuatro modelos de fricción distintos.

Además del modelo lineal simétrico, se ha considerado el modelo de fricción propuesto por [4], en el que se asume un comportamiento asimétrico. El tercer modelo utilizado permite un mejor ajuste al par de fricción real mediante la inclusión de un tercer término no lineal respecto a la velocidad [3]. El último modelo considera la forma asimétrica del modelo anterior.

Generalizando, el sistema de ecuaciones correspondiente al modelo de fricción expresado en su forma matricial presentará, por tanto, la siguiente forma para una configuración $(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}})$ del robot dada:

$$\bar{\tau}_f = \mathbf{W}_f(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi}_f \quad (10)$$

donde $\bar{\tau}_f$ es el vector de fuerzas generalizadas, \mathbf{W}_f

la matriz de observación y $\bar{\Phi}_f$ es el vector de parámetros del sistema, en este caso los coeficientes del modelo de fricción, el cual presenta la siguiente forma:

$$\bar{\Phi}_f = [F_{n_1} \dots F_{n_m} \dots F_{l_1} \dots F_{l_m}]^T$$

A partir de los modelos de elementos, actuadores y fricción comentados anteriormente se puede obtener el modelo dinámico completo del robot manipulador de acuerdo a las ecuaciones de Gibbs-Appell;

$$\begin{aligned} \tau_k &= \tau_{l_k} + \tau_{r_k} + \tau_{f_k} = \\ &= \underbrace{\sum_{i=1}^n \frac{\partial G_{l_i}}{\partial \ddot{q}_k}}_{\text{barras}} + \underbrace{\sum_{i=1}^n \frac{\partial G_{r_i}}{\partial \ddot{q}_k}}_{\text{rotores}} + \underbrace{F_{f_k}(\dot{q}_k)}_{\text{fricción}} \end{aligned} \quad (11)$$

Tras desarrollar los términos de la anterior expresión, una que vez las ecuaciones han sido linealizadas y puestas en su forma matricial, tal como se describirá en los siguientes apartados, ésta quedará de la siguiente forma:

$$\bar{\tau} = \mathbf{W}(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi} = \begin{bmatrix} \mathbf{W}_l & | & \mathbf{W}_r & | & \mathbf{W}_f \end{bmatrix} \cdot \begin{bmatrix} \bar{\Phi}_l \\ \bar{\Phi}_r \\ \bar{\Phi}_f \end{bmatrix} \quad (12)$$

3 IDENTIFICACIÓN DEL MODELO DINÁMICO DE ROBOTS MANIPULADORES

Para poder establecer el control dinámico del robot manipulador es necesario abordar la identificación de los distintos parámetros dinámicos de las ecuaciones dinámicas. En este trabajo se han planteado e implementado dos métodos de identificación: un método de identificación directo y una variante del esquema de identificación indirecto.

El método de identificación directo se basa esencialmente en [2], aunque en este caso las trayectorias optimizadas utilizadas para obtener la identificación han sido parametrizadas mediante series finitas de Fourier, tal como se describe en [7] y [8]. Éste constituye quizás el esquema de identificación más comúnmente empleado en el caso de robots manipuladores de cadena cinemática abierta, lo cual, unido a las ventajas que presenta frente a los esquemas de identificación indirectos, hacen de él un método muy conveniente para obtener la identificación paramétrica.

En cuanto al segundo método, éste puede considerarse como una combinación de los esquemas de identificación directo e indirecto. En este caso se ha prescindido de implementar un método de identificación enteramente de acuerdo al esquema de identificación indirecta por considerar su aplicación práctica costosa y poco sistemática, requiriendo de un diseño específico para cada tipo de robot.

Con el método de identificación directo, el modelo dinámico del robot manipulador es identificado en su totalidad a partir de un solo experimento. En la Figura 1 puede verse un esquema general del método tal y como se ha implementado.

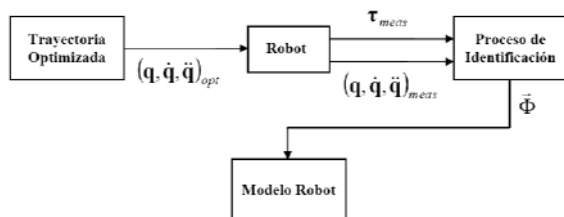


Figura 1. Esquema de identificación directa

Con este método, partiendo de las ecuaciones del movimiento en forma lineal desarrolladas en el apartado anterior, se obtiene una matriz de observación \mathbf{W}_i para cada una de las configuraciones del brazo robot consideradas $(\bar{q}_i, \dot{\bar{q}}_i, \ddot{\bar{q}}_i)$, con $i=1\dots m$. Dichas matrices estarán constituidas por tantas filas como grados de libertad posea el robot y tantas columnas como parámetros del modelo dinámico a ser identificados.

$$\bar{\tau}_i = \mathbf{W}_i \cdot \bar{\Phi} \quad (13)$$

El sistema de ecuaciones de la expresión (13) puede ser reducido a un sistema de ecuaciones de rango completo, donde tan sólo aparecen aquellos parámetros base del modelo que determinan el comportamiento dinámico del mismo de forma independiente. Con ello, se llega al sistema de ecuaciones reducido:

$$\bar{\tau}_i = \mathbf{W}_{B_i} \cdot \bar{\Phi}_B \quad (14)$$

A continuación se deben agregar las matrices correspondientes al conjunto de configuraciones considerado para el proceso de identificación. La matriz resultante constituye la matriz de observación del sistema para dicho conjunto de configuraciones. Del mismo modo, se construye un vector de pares en los nudos a partir de los vectores de pares en los nudos correspondientes a cada una de las configuraciones consideradas. Así pues, se obtiene el sistema de ecuaciones definido por la siguiente expresión:

$${}^a\bar{\tau} = {}^a\mathbf{W}_B \cdot \bar{\Phi}_B \quad (15)$$

donde:

$${}^a\bar{\tau} = \begin{bmatrix} \bar{\tau}_1 \\ \vdots \\ \bar{\tau}_m \end{bmatrix} \quad {}^a\mathbf{W}_B = \begin{bmatrix} {}^a\mathbf{W}_{B1} \\ \vdots \\ {}^a\mathbf{W}_{Bm} \end{bmatrix}$$

El número m de configuraciones debe ser elegido de tal modo que el número de filas de la matriz de observación sea mayor que el número de columnas, a fin de obtener un sistema de ecuaciones sobredeterminado. Dicho sistema lineal de ecuaciones será resuelto mediante la aplicación del LSM.

Con la finalidad de minimizar la influencia de los errores de medida sobre los parámetros identificados, la matriz de observación obtenida debería presentar un número de condición suficientemente reducido, así como un adecuado equilibrio numérico de sus elementos. El número de condición de la matriz de observación viene determinado por la trayectoria aplicada para realizar la identificación y, por consiguiente, solamente puede ser mejorado a través

de un diseño conveniente de dicha trayectoria a fin de que ésta logre excitar suficientemente los parámetros a identificar.

Como se ha comentado, el segundo método de identificación desarrollado es una variante propuesta del método indirecto para la identificación del modelo dinámico de un robot manipulador. Ésta se lleva a cabo en dos pasos: Primero se realiza un experimento a partir del cual se estiman los parámetros de gravedad del modelo dinámico y los coeficientes del modelo de fricción. En el segundo paso, se obtienen datos de posición angular y par en los nudos de un experimento en el que se considera una trayectoria optimizada parametrizada mediante series finitas de Fourier. Las medidas de par en los nudos son compensadas mediante los términos de gravedad y fricción ya identificados del modelo dinámico con el fin de identificar el resto del modelo. En la Figura 2 puede verse un diagrama en el que se representa el esquema de identificación indirecta aplicado.

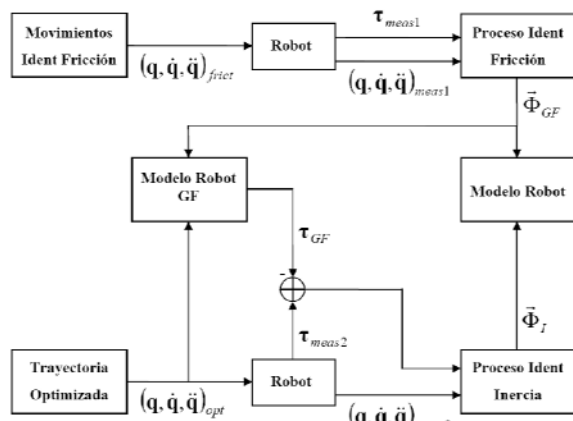


Figura 2: Esquema de identificación indirecta.

El sistema de ecuaciones planteado para la identificación directa obtenido a partir de las ecuaciones de la dinámica en forma lineal puede ser descompuesto en tres partes según la naturaleza del efecto dinámico implicado en cada caso de acuerdo a la siguiente expresión:

$$\bar{\tau}_i = \mathbf{W}_{Bi} \cdot \bar{\Phi}_B \equiv \begin{bmatrix} \bar{\tau}_{Ii} \\ \bar{\tau}_{Gi} \\ \bar{\tau}_{Fi} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{BIi} & \mathbf{W}_{BGi} & \mathbf{W}_{BFi} \end{bmatrix} \begin{bmatrix} \bar{\Phi}_{BI} \\ \bar{\Phi}_{BG} \\ \bar{\Phi}_{BF} \end{bmatrix} \quad (16)$$

donde $\bar{\Phi}_{BI}$, $\bar{\Phi}_{BG}$ y $\bar{\Phi}_{BF}$ son, respectivamente, los parámetros base asociados a los elementos de la matriz de inercia, los parámetros gravitatorios y los de fricción. A su vez, \mathbf{W}_{BIi} , \mathbf{W}_{BGi} , \mathbf{W}_{BFi} y $\bar{\tau}_{Ii}$, $\bar{\tau}_{Gi}$, $\bar{\tau}_{Fi}$ son los términos correspondientes de la matriz de observación y del vector de pares en los nudos del sistema. Además, los parámetros de fricción son los coeficientes del modelo de fricción aplicado.

En el trabajo se han aplicado los modelos dinámicos de elementos, de elementos y actuadores, así como tres modelos de fricción distintos a fin de llevar a cabo el procedimiento de identificación indirecta propuesto. La utilización de modelos de fricción asimétricos implica una dificultad adicional para la identificación del propio modelo de fricción debido al hecho de que las contribuciones al par en los nudos correspondientes a los fenómenos de fricción y a la gravedad no pueden ser separadas. Como solución a este inconveniente, la identificación de los parámetros gravitatorios se realiza en el primer paso del proceso de identificación, conjuntamente con los parámetros de fricción.

La identificación de los coeficientes de fricción correspondientes a cada articulación se realiza secuencialmente empezando por el último nudo hasta llegar al primero. En cada paso, solamente una articulación es movida a velocidad constante, en tanto que el resto permanece en reposo en una configuración determinada. Los parámetros gravitatorios que afectan el movimiento de la articulación correspondiente son igualmente identificados junto con los coeficientes de fricción. A partir de un conjunto de configuraciones del manipulador, se construye una matriz de observación con información acerca de la gravedad y la fricción, así como un vector con las medidas de pares en los nudos, siendo el sistema lineal de ecuaciones así constituido posteriormente resuelto por medio de un algoritmo de LSM estándar.

Una explicación más detallada de los métodos de identificación desarrollados y un análisis comparativo de los resultados obtenidos se puede encontrar en [1] y [5].

4 VALIDACIÓN IDENTIFICACIÓN PARAMÉTRICA. CONTROL EN TIEMPO REAL DE ROBOTS MANIPULADORES

Para poder verificar los distintos procedimientos desarrollados en este trabajo se han llevado a cabo diversos experimentos de identificación utilizando tanto datos simulados como datos medidos de un robot manipulador industrial: el PUMA560. La Figura 3 muestra el esquema del procedimiento experimental seguido que permite contrastar la precisión del modelo dinámico propuesto con respecto a las medidas del par en las articulaciones tomadas del robot real.

La trayectoria aplicada para llevar a cabo la identificación de los parámetros ha sido parametrizada mediante series finitas de Fourier. En

cuanto al procedimiento de optimización, se ha tomado el número de condición de la matriz de observación del sistema como criterio de optimización y se han impuesto las restricciones cinemáticas pertinentes tanto en el espacio de articulaciones como en el espacio cartesiano de trabajo relativas al robot y a su entorno de trabajo. Se ha aplicado un algoritmo de optimización basado en SQP.

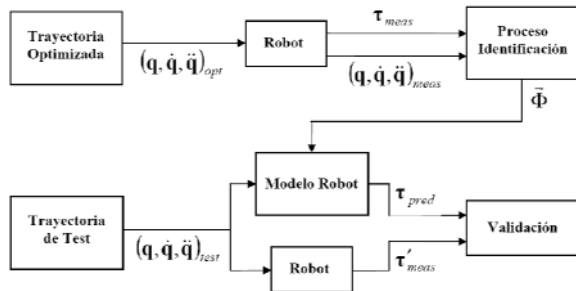


Figura 3: Esquema de validación utilizando un robot manipulador industrial

Una vez que la matriz de observación ha sido calculada a partir de los datos de posición angular y par en los nudos, el proceso de identificación prosigue tal como se ha descrito anteriormente de acuerdo al esquema de identificación que se esté aplicando en cada caso.

La Figura 4 muestra los pares obtenidos de los 6 grados de libertad del PUMA. En línea continua se muestran los pares medidos en las articulaciones del robot real, y en la línea discontinua se tienen los pares calculados por el modelo del robot con los parámetros identificados. Tal como se puede observar, las diferencias entre los pares de cada una de las articulaciones son prácticamente iguales.

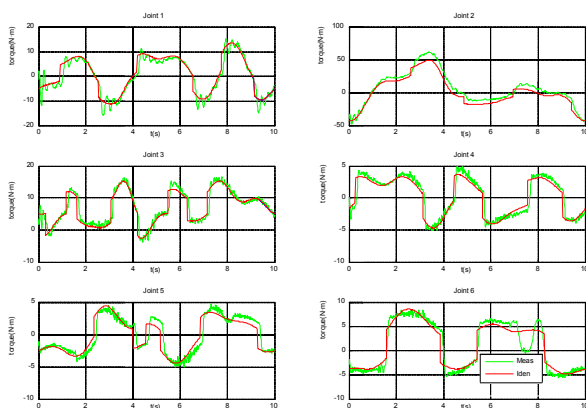


Figura 4: Test del procedimiento de identificación del conjunto de parámetros base del PUMA 560

Para poder llevar a cabo la identificación y el control del robot PUMA 560 se propuso la sustitución de la unidad de control original del sistema mediante una

nueva basada en un computador personal industrial. De este modo, el PC accede a las posiciones angulares de los nudos y genera las acciones de control requeridas para accionar los distintos elementos del brazo robot. La Figura 5 muestra el robot utilizado en este trabajo.



Figura 5: Robot industrial PUMA 560

Como puede observarse en la Figura 6, las lecturas de las posiciones angulares de los nudos del brazo robot son obtenidas a través de las tarjetas de adquisición de datos, la cuales recogen las señales de los potenciómetros y los encoders incrementales. Para ello se utilizan 2 clases de tarjetas de Advantech: la PCI-1711 para la lectura de los potenciómetros y la PCL-833 para la lectura de los encoders.

Con la información del estado del sistema, las referencias de movimiento y el algoritmo de control programados, el PC es capaz de calcular las acciones de control necesarias para las distintas articulaciones del robot. Estas acciones de control se suministran a los amplificadores de potencia del controlador mediante los convertidores Digital/Analógico disponibles es las tarjetas PCI-1720 de Advantech.

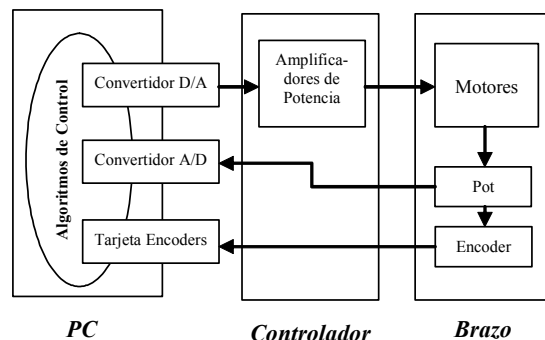


Figura 6: Arquitectura de Control del PUMA 560

Una información más detallada de la arquitectura hardware/software desarrollada para el control de este robot se puede encontrar en [10]

Una vez obtenidos y validados los parámetros de la ecuación dinámica del robot se puede programar cualquier controlador dinámico. Aunque se han desarrollado otros controles (por ejemplo los basados en Pasividad), en este trabajo se ha utilizado una estrategia de control por par calculado (o dinámica inversa) que permite establecer el control de trayectoria. La estrategia de control viene dada por la expresión siguiente:

$$\tau_c = M(q, \theta)v + C(q, \dot{q}, \theta)\dot{q} + G(q, \theta) \quad (17)$$

donde:

τ_c : par de control

$M(q, \theta)$: matriz de inercias del sistema

$C(q, \dot{q}, \theta)\dot{q}$: vector Coriolis y fuerzas centrífugas

$G(q, \theta)$: vector de fuerzas gravitatorias

La matriz de inercias $M(q, \theta)$ se puede obtener directamente a partir de los términos B_k de la ecuación de movimiento de Gibbs-Appell de la expresión (5). De la misma forma, los términos correspondientes a las fuerzas de Coriolis, centrífugas y gravitatorias $C(q, \dot{q})\dot{q} + G(q)$ se pueden calcular directamente a partir de los términos A_k de la ecuación (5). Una información más detallada sobre la obtención de estos términos se puede encontrar en [9].

Es interesante añadir que, dependiendo de la expresión del término v , se pueden obtener diferentes estrategias de control tal como se muestra en la tabla siguiente:

Tipo de Controlador	v
Punto a Punto	$-K_d\dot{q} - K_p e$
Control de Trayectoria	$\ddot{q}_d - K_d\dot{e} - K_p e$
Control de Trayectoria con acción integral	$\ddot{q}_d - K_d\dot{q} - K_p e - K_i \int_0^t e(u) du$

Tabla 1. Controladores por dinámica inversa

donde:

$$e = q - q_d$$

K_d , K_p y K_i : matrices (diagonales y definidas positivas) de ganancias derivativas, proporcionales e integradoras

La nueva arquitectura abierta de robot permite implementar el control de los 6 grados de libertad del PUMA con cualquier estrategia de control. A continuación se muestra la respuesta de los 3 primeros ejes del robot real cuando se utiliza un controlador de trayectoria con acción integral por dinámica inversa.

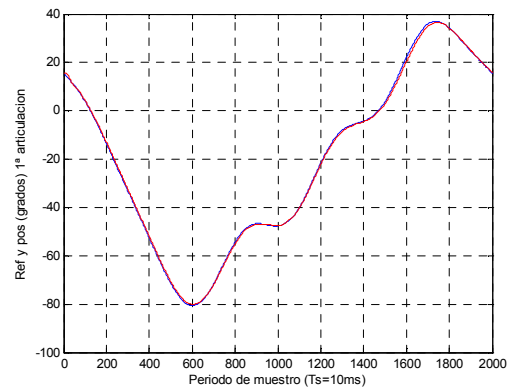


Figura 7: Respuesta 1ª articulación PUMA 560

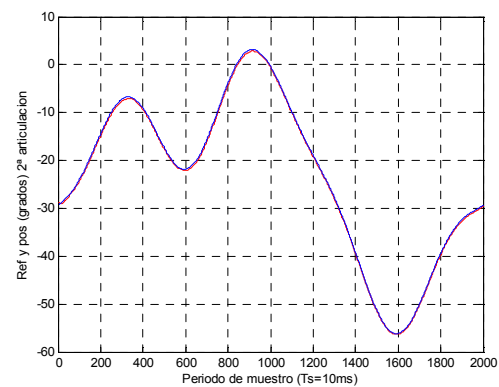


Figura 8: Respuesta 2ª articulación PUMA 560

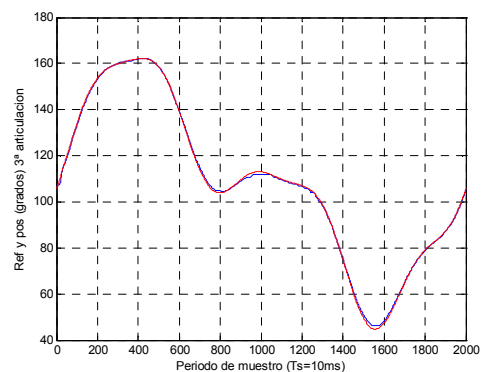


Figura 9: Respuesta 3ª articulación PUMA 560

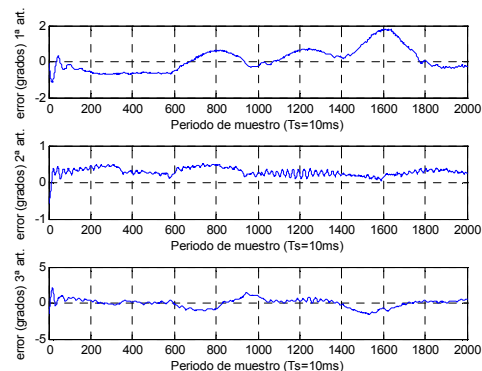


Figura 10: Señal de error de las 3 articulaciones

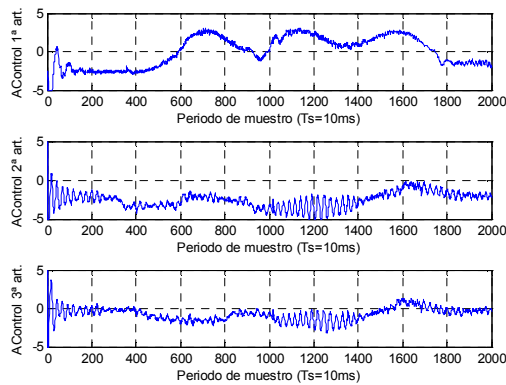


Figura 11: Acciones de control suministradas

Tal como se puede apreciar, el comportamiento del robot es muy bueno, siguiendo las referencias especificadas con un error casi despreciable.

5 CONCLUSIONES

En el artículo se ha abordado la problemática de la identificación experimental de los parámetros que definen el modelo dinámico de un robot manipulador industrial. Se ha empleado un modelo dinámico de elementos rígidos del brazo robot, el cual se ha completado mediante el modelado de los actuadores, a fin de permitir la consideración del efecto giroscópico de los rotores. El modelo dinámico del robot se ha desarrollado a partir del formalismo dinámico de Gibbs-Appell, el cual aporta distintas ventajas, entre ellas la sencillez de las expresiones obtenidas y la fácil integración de la dinámica de los actuadores en el modelo. Las ecuaciones del modelo dinámico obtenidas han sido expresadas linealmente con respecto a los parámetros a identificar y expresadas en forma matricial. Por otra parte, se ha modelado asimismo el par debido a la fricción en las articulaciones, para lo cual se han considerado cuatro modelos distintos.

Por lo que respecta a la identificación de los parámetros dinámicos, se han implementado tanto un método basado en el esquema de identificación directo como un método parcialmente basado en el esquema de identificación indirecto, mediante el cual se ha tratado de sacar provecho de algunas ventajas inherentes a cada uno de los esquemas de identificación mencionados.

Finalmente, tanto el modelado como los procedimientos de identificación paramétrica se han verificado en un sistema real: un robot industrial PUMA 560, mostrando la respuesta de dicho robot con un control de trayectoria basado en la dinámica inversa.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos FEDER-CICYT con número de referencia DPI2005-08732-C02-01 y DPI2005-08732-C02-02, del Ministerio de Educación y Ciencia.

Referencias

- [1] F. Benimeli, V. Mata, and F. Valero, "A Comparison between Direct and Mixed Dynamic Parameter Identification Methods in Mechanical Systems," presented at MULTIBODY DYNAMICS 2005, ECCOMAS Thematic Conference, Madrid, Spain, 2005.
- [2] M. Gautier, "Numerical-Calculation of the Base Inertial Parameters of Robots," *Journal of Robotic Systems*, vol. 8, pp. 485-506, 1991.
- [3] M. Grotjahn, M. Daemi, and B. Heimann, "Friction and rigid body identification of robot dynamics," *International Journal of Solids and Structures*, vol. 38, pp. 1889-1902, 2001.
- [4] Z. R. Lu, K. B. Shimoga, and A. A. Goldenberg, "Experimental-Determination of Dynamic Parameters of Robotic Arms," *Journal of Robotic Systems*, vol. 10, pp. 1009-1029, 1993.
- [5] V. Mata, F. Benimeli, N. Farhat, and A. Valera, "Dynamic Parameter Identification in Industrial Robots Considering Physical Feasibility," *Advanced Robotics*, vol. 19, pp. 101-119, 2005.
- [6] S. Provenzano, "Aplicación de las Ecuaciones de Gibbs-Appell a la Dinámica de Robots," in, ed. Valencia: Universitat Politècnica de Valencia, 2001.
- [7] J. Swevers, C. Ganseman, J. DeSchutter, and H. VanBrussel, "Experimental robot identification using optimised periodic trajectories," vol. 10, pp. 561-577, 1996.
- [8] J. Swevers, C. Ganseman, D. B. Tukul, J. DeSchutter, and H. VanBrussel, "Optimal robot excitation and identification," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 730-740, 1997.
- [9] A. Valera, D. Jornet, V. Mata, F. Valero, "Implementation of the Inverse Dynamics Control for a PUMA 560 using a New Open Architecture," 15th ISPE/IEE International Conference on CAD/CAM, Robotics & Factories of the Future, Campinas (Brasil), 1999.
- [10] A. Valera, V. Mata, M. Valles, F. Valero, N. Rosillo, and F. Benimeli, "Solving the Inverse Dynamics Control for Low Cost Real-Time Industrial Robot Control Applications," *Robotica*, vol. 21, pp. 261-269, 2003.

DETECCIÓN DE FALLOS EN EL PROCESO DE TERMOSELLADO DE TARRINAS PARA PRODUCTOS ALIMENTICIOS

Antonio José Calderón Godoy
Universidad de Extremadura, ajcalde@unex.es

Resumen

En este trabajo se aborda el diseño y desarrollo de un sistema automático de detección de defectos en tiempo real en el sellado de alimentos envasados, herméticamente cerrados. Este sistema irá ubicado justo después de la salida de la etapa de termosellado y antes del embalaje y paletizado de los envases.

Palabras Clave: Detección de fallos, termosellado.

1 INTRODUCCIÓN

El sellado de los alimentos envasados y termosellados, herméticamente cerrados, debe realizarse correctamente para garantizar que el alimento podrá mantener sus características inalteradas durante un periodo de tiempo prolongado.

Existen una serie de **factores** que influyen en el proceso de termosellado y que han de tenerse en cuenta a fin de conseguir un sellado hermético. Estos factores pueden agruparse en dos categorías:

- Características de los materiales (composición y diseño)
- Tecnología aplicada (equipo y condiciones)

Dentro de estas categorías, los parámetros más importantes a considerar serán: porosidad del material, superficies de contacto, adhesión, solapado, etc. y las condiciones de presión y temperatura que definirán la continuidad del cierre.

Actualmente los procesos más importantes empleados para la verificación del sellado de alimentos envasados y termosellados, herméticamente cerrados, son la inspección visual y la detección de microfugas.

La **inspección visual** es un proceso no destructivo realizado a través de planes de muestreo, basados en procedimientos estadísticos, diferentes según la índole de la inspección que se desea realizar. Consiste en realizar una comprobación del cierre de determinado número de envases tomados al final de

la cadena. Sin embargo, no son examinados el 100% de ellos y por tanto hay riesgo de dar por válidos envases que no estén herméticamente cerrados y que, por tanto, sean no aceptables pero no detectados.

Este método por sí solo no proporciona un alto nivel de seguridad para detectar envases defectuosos. Sólo proporcionará una idea de la aceptabilidad e inocuidad de los envases sospechosos examinados y permitirá prever la contaminación posterior.

Como ventaja podemos decir que permite examinar un gran número de envases y el coste es mínimo, pero existen limitaciones en cuanto a los recursos y representatividad de la muestra; no siendo todos los defectos y/o fugas detectables. Por lo tanto, este método debe ir acompañado de otras pruebas discriminatorias como: controles de hermeticidad, fuerza y continuidad del cierre.

Por otra parte podemos citar determinados métodos de **detección de fugas** (ensayo de aire a presión, del Helio, del Sulfuro de Hidrógeno, etc.) empleados en combinación con los ensayos microbiológicos, para corroborar los resultados obtenidos en el laboratorio.

También tienen sus ventajas (rapidez, magnitud de la fuga, proporcionan dato permanente) y desventajas (no indican el punto de fuga, difícil selección del ensayo más adecuado, etc.)

Por lo tanto, los sistemas existentes actualmente para la detección de defectos de sellado de la tapa (colocada en el proceso de cierre de tarrinas) adolecen de una serie de limitaciones que los hacen inapropiados para su aplicación directa en el proceso productivo.

Entre estas limitaciones se pueden destacar el elevado precio de estos sistemas, que lo hacen inaplicable en productos de bajo valor comercial unitario, y su elevada cadencia, lo que impide que se pueda verificar en tiempo real el total de la producción. La solución adoptada para paliar este último inconveniente es realizar un muestreo de los envases cerrados, lo que no garantiza una fiabilidad del 100% en el producto de salida [1-6].

Por otra parte, la etapa de cierre de envases es la última antes del embalaje y paletizado de los mismos para su expedición. Por ello, un defecto en el sellado (cierre) no será detectado hasta que el producto no llegue al punto de venta o, incluso, al consumidor.

El resto del artículo está estructurado como sigue. En la sección 2 se describe el sistema que se pretende desarrollar para la detección de fallos. En la sección 3 se expone el funcionamiento del sistema descrito. En la sección 4 se realizan algunas consideraciones sobre las necesidades para implantación del prototipo desarrollado en la cadena de producción en la planta real. Finalmente, en la sección 5 se presentan las conclusiones y pretensiones de este trabajo.

2 DESCRIPCIÓN DEL SISTEMA

En este proyecto se propone un sistema para la detección en tiempo real de defectos en el sellado (cierre) de las tapas de envases (tarrinas de compota). En la figura 1 aparece una imagen del tipo de envase a inspeccionar. En ella se puede apreciar la morfología de la tarrina, siendo las medidas a tener en cuenta para el diseño del sistema de detección de fallos las siguientes: altura = 33 mm, diámetro = 84 mm.



Figura 1: Tarrina de compota a inspeccionar

Ante la falta de métodos en continuo, no excesivamente caros y suficientemente eficaces, se considera justificada la necesidad de buscar un método de detección de fugas y verificación del sellado, para instalar al final de las líneas de producción, de aplicabilidad a este tipo de alimentos envasados, siendo examinada la totalidad de la producción antes de ser introducida en el embalaje.

Con este proyecto se trata de desarrollar estrategias de detección, inspección y control automático de fallos de sellado, con una garantía de ausencia de éstos sobre el 100% de la producción.

Se ha construido una plataforma experimental donde ensayar la estrategia propuesta. Esta plataforma debe,

a su vez, servir como prototipo del sistema a instalar al final de las cadenas de producción que incluyan líneas de sellado de productos envasados, herméticamente cerrados. Además, en los que se deba garantizar la inspección sobre toda la producción. Estos condicionantes obligan a que el sistema desarrollado tenga la capacidad de funcionar en ambiente industrial y trabaje en tiempo real, no suponiendo una merma en la capacidad productiva de la línea de producción en la que se instale.

Este sistema estará formado por un alimentador de tarrinas procedentes de la máquina de termosellado, una estación para la detección de envases con defectos de sellado y un sistema de identificación y expulsión de tarrina defectuosa. Todas estas estaciones estarán conectadas mediante una cinta transportadora (conveyor) por la que discurrirán las tarrinas (envases).

El conjunto resultante puede apreciarse en la representación esquemática que aparece en la figura 2. Cada una de las estaciones estará constituida por una estructura construida en perfilaría de aluminio, sobre la cual se sitúan los componentes encargados de realizar el proceso correspondiente, incluyendo las electroválvulas de control cuando se trate de componentes neumáticos.

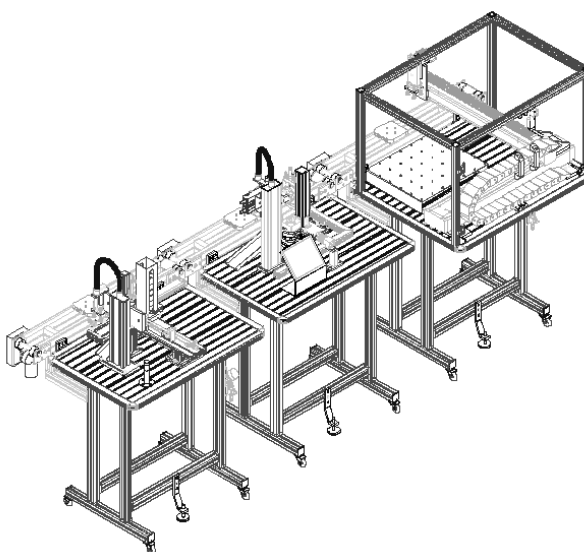


Figura 2: Representación esquemática del sistema completo

2.1 ESTACIÓN DE DETECCIÓN DE DEFECTOS

Para la detección de fallos o defectos en el sellado se han de utilizar métodos que no afecten (alteren) el contenido de las tarrinas, a la vez que garantizan una frecuencia de procesado de tarrinas suficiente para la inspección de todas y cada una de ellas.

En el momento en que se planteó el proyecto, el llenado de las tarrinas se llevaba a cabo con el producto en frío. Esto traía como consecuencia que al cerrar la tarrina se producía una leve sobrepresión en el interior del envase que provocaba un ligero abombamiento en la tapa. Por tal motivo se decidió actuar justo después de la estación de termosellado para la detección de los defectos del mismo.

El sistema de detección propuesto consistía en aplicar presión con un palpador sobre la tapa y medir el desplazamiento producido cuando existe anomalía o defecto en el sellado (cierre). Para esta tarea se utilizó el sensor **SMAT-8E-S50-IU-M8**. El sistema ideado respondía al esquema mostrado en la figura 3.

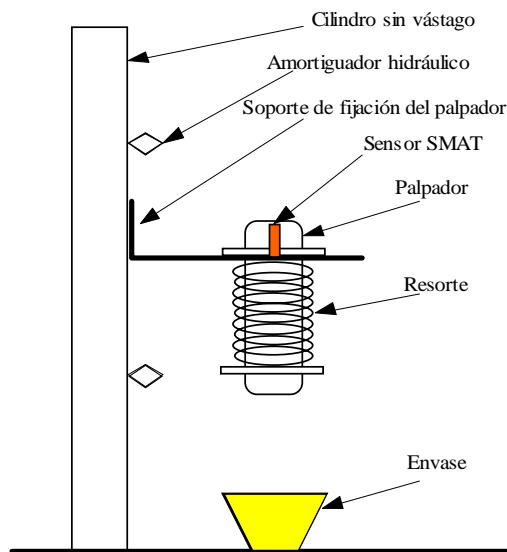


Figura 3: Representación esquemática del sistema propuesto

Como se puede apreciar, el sistema palpador-resorte va montado sobre un soporte de aluminio. El sensor de desplazamiento SMAT está fijado solidariamente a este soporte, de manera que puede medir el desplazamiento relativo producido entre el soporte y el palpador.

El movimiento en sentido vertical de este soporte (arriba/abajo) se consigue utilizando un actuador lineal sin vástago del tipo **DGC-G** de 12 mm de Ø. Para el control del cilindro se utiliza un distribuidor electroneumático de 5 vías/2 posiciones, gobernado por las salidas digitales de un autómata programable **s7_224**. Las posiciones superior e inferior del sistema soporte-palpador están fijadas por sendos amortiguadores hidráulicos. La ubicación del amortiguador inferior está determinada por la máxima presión que el palpador debe ejercer sobre la tarrina. Esta presión se determina experimentalmente, de forma que no se dañe la tapa del envase, pero produzca una deformación suficiente en el resorte

que permita la detección de fugas. Es precisamente la aparición de estas fugas el facto que se utilizará para determinar la existencia de defectos en el termosellado.

3 FUNCIONAMIENTO DE LA ESTACIÓN DE DETECCIÓN DE FALLOS

El funcionamiento del sistema descrito es como sigue. El sistema palpador bajará hasta una posición predeterminada, limitada por amortiguadores hidráulicos, de forma que se produzca una ligera deflexión del resorte. De esta forma, el palpador realizará una leve presión sobre la tapa del envase a ensayar. Si el sellado no presenta fisuras, una vez alcanzado el equilibrio, el palpador no se desplazará. Sin embargo, si existieran defectos de sellado, éstos provocarían fugas que evitarán que el sistema palpador alcance el equilibrio en el tiempo determinado y siga desplazándose por efecto del resorte. La medida de este desplazamiento se puede utilizar para discriminar los envases sellados correctamente de los que presentan defectos en el proceso de termosellado.

Tanto el control del movimiento del actuador lineal como la medida del desplazamiento se lleva a cabo mediante el autómata programable **s7_224** de Siemens, al que se ha dotado del módulo de 4 entradas y una salida analógicas **EM235**. Este módulo de entradas/salidas analógicas tiene una resolución de 12 bits cuando está configurado para trabajar con señales unipolares, como es nuestro caso. Esta resolución es suficiente para la aplicación y permite aprovechar las características de definición del sensor.

El sistema se completará con una estación de alimentación de los envases procedentes de la máquina de termosellado, una estación para la inspección de envases con detección de defectos de sellado (la descrita) y una estación de almacenaje de envases inspeccionados, que hará las veces de almacén de la fase de expedición. Todas estas estaciones estarán conectadas mediante un sistema de transporte. Este sistema de transporte está compuesto por un transfer acoplado a cada estación de forma que se permita la configuración de varios Lay-out, con la posibilidad de ampliar el sistema para incrementar las capacidades del mismo.

4 RESULTADOS PREVIOS

El prototipo descrito se encuentra actualmente en fase de desarrollo. La estación de detección de fallos se ha probado experimentalmente en laboratorio y se

está adaptando a las exigencias de la cadena de producción real.

Se han testado envases sin defectos, para ajustar el funcionamiento de la estación cuando el termosellado no presenta fisuras. En tal caso, la deformación de la tapa debida a la presión del palpador estaba alrededor de 1 mm. La excursión máxima del conjunto soporte-palpador se ha limitado a 3 mm una vez que se alcanza la posición del envase. Esto es debido a que, en el caso de un defecto grave de sellado, si el palpador siguiera avanzando provocaría el derrame del producto envasado.

También se ha de ajustar el tiempo empleado en llevar a cabo el procedimiento completo: ubicación del envase en la estación, desplazamiento del conjunto soporte-palpador y tiempo de equilibrio. Para obtener el límite superior de este tiempo se ha de considerar la cadencia de salida de envases de la estación de termosellado. Como la cadencia de salida está comprendida entre 900 y 1.200 envases por hora, esto proporciona una tasa de 15 a 20 envases por minuto. Así se obtiene un tiempo límite para realizar todas las operaciones de:

$$\text{Tiempo máximo de operación} = \frac{60}{20} = 3 \text{ seg.}$$

Considerando este tiempo máximo para realizar todas las operaciones, el tiempo asignado para alcanzar el equilibrio entre la deformación del resorte y la sobrepresión en el interior del envase es de 1 seg. Si transcurrido este tiempo, el soporte siguiera descendiendo, esto supondría la existencia de fugas en el cierre y por lo tanto un defecto en el termosellado de la tarrina.

La tasa de defectos observada en la planta actual, con detección por mera inspección visual de operarios al final de la cadena de producción está en torno al 3%, por lo tanto se justifica la instalación de este sistema de detección de fallos, antes de la expedición del producto terminado para consumo.

5 CONCLUSIONES

En este trabajo se ha presentado una propuesta basada en la sobrepresión producida en interior de los envases con llenado de producto en frío para la detección automática del termosellado de los mismos. Se han descrito los componentes hardware del sistema, así como el principio de funcionamiento del prototipo desarrollado para automatizar la detección de envases con defectos. El periodo de caducidad de los productos envasados en frío está en torno a los tres meses. En el momento actual se está

trabajando con la posibilidad de llenar el producto en caliente, lo que alargaría el periodo de caducidad a doce meses. Este cambio traerá como consecuencia una redefinición completa del sistema de detección automática de fallos de termosellado. De este modo, el marco en el cual se desarrollará el trabajo futuro tendrá en consideración esta modificación en el proceso productivo.

Agradecimientos

Este trabajo está parcialmente financiado por el proyecto de desarrollo tecnológico PDT05A49 de la Junta de Extremadura.

Referencias (10 ptos, negrita)

- [1] AENOR-.CERNA (1982)
- [2] Buckle (1985)
- [3] Código internacional recomendado de prácticas de higiene para alimentos poco ácidos y alimentos acidificados envasados. CAC/RCP 23-1979, Rev.2
- [4] Directrices sobre procedimientos básicos para la inspección visual de lotes de alimentos envasados. CAC/ GL 17- 1993.
- [5] U.S.F.D.A (1984)
- [6] www.codexalimentarius.net

ENSEÑANZA Y APRENDIZAJE DE CONCEPTOS DE CONTROL AUTOMÁTICO UTILIZANDO LABVIEW

J.L. Guzmán, F. Rodríguez, M. Berenguel, D. Lacasa

Universidad de Almería. Escuela Politécnica Superior. Dpto. de Lenguajes y Computación.

Ctra. Sacramento s/n, 04120 Almería

[joguzman@ual.es, frrodrig@ual.es, beren@ual.es, dlacasa@ual.es]

Resumen

Este trabajo describe un conjunto de sesiones prácticas de distintos estudios de ingeniería en la Universidad de Almería donde se utiliza LabVIEW como herramienta docente con el fin de introducir conceptos básicos de control automático. El uso de LabVIEW en las sesiones prácticas esta presente en aplicaciones desarrolladas por el profesorado que son utilizadas por los estudiantes, así como el uso de LabVIEW como entorno de programación por parte del alumnado. Las principales tareas en las que se centran las diferentes sesiones son: modelado, control monovariable y multivariable, e implementación de herramientas SCADA.

Palabras Clave: educación en automática, modelado, control automático, instrumentación virtual.

1 INTRODUCCIÓN

Los avances docentes producidos en los últimos años dentro del área de Ingeniería de Sistemas y Automática se han visto claramente reflejados en numerosos trabajos y publicaciones principalmente relacionados con laboratorios virtuales ([9], [13] [16]), laboratorios remotos ([3], [4], [8], [17], [18]) y herramientas interactivas ([2], [5], [6], [7]). La investigación dedicada a la búsqueda de nuevas tecnologías y métodos docentes parecía algo que siempre se encontraba en un segundo plano por parte de la comunidad universitaria, pero sin embargo, se puede observar cómo actualmente existe una apuesta firme en este ámbito donde se establece una línea de investigación al mismo nivel que otras áreas de interés científico. Un buen ejemplo de ello se puede observar en IFAC y en IEEECS, donde en ambas asociaciones se encuentra presente un área técnica denominada *Educación en Control* (Control Education) soportada por dos comités científico-técnicos y estando al mismo nivel que el resto de áreas técnicas [5].

El hecho de esta dedicación a la investigación docente en el ámbito de *Automática*, se debe fundamentalmente a la fuerte componente práctica que envuelve la enseñanza y aprendizaje de esta disciplina. Existen ciertas áreas donde la docencia se centra fundamentalmente en clases teóricas apoyadas por una serie de ejercicios teórico-prácticos. Sin embargo, dentro del campo de la ingeniería la enseñanza debe ir un poco más allá, donde los estudiantes puedan poner en práctica los conocimientos teóricos trabajando con sistemas reales o simuladores que reflejen lo más fielmente la realidad [2].

Los aspectos teóricos se pueden aprender de los libros, de inspiradas lecciones y de un activo estudio, pero sin embargo, la habilidad de resolver problemas prácticos subyace en la destreza de cómo utilizar dicha teoría para poder trasladar problemas de gran magnitud en problemas sencillos de resolver [2]. Una de las tareas más importantes por parte de los profesores en ingeniería consiste en transmitir a los estudiantes no solamente la parte formal de la disciplina, sino también y con mayor énfasis, la estrategia y aspectos intuitivos de la materia.

Un elemento importante de la docencia en Automática es que se centra en el continuo uso de experimentos y desarrollo de nuevos laboratorios y herramientas basadas en computador. Este tipo de herramientas deberían ser integradas como parte de los contenidos docentes, no sólo a nivel de cursos introductorios como actualmente están presentes, sino también para aumentar su difusión y uso en trabajos de cursos avanzados de segundo o tercer ciclo [2], [5].

En este trabajo se presenta *LabVIEW* de National Instruments [11], [12], como una herramienta de gran potencial que permite poder llevar a cabo las tareas docentes comentadas anteriormente. Se describirán un conjunto de herramientas que han sido desarrolladas por el profesorado del área de Ingeniería de Sistemas y Automática de la

Universidad de Almería, y que tienen como fin el servir de apoyo para la enseñanza y aprendizaje de conceptos de control automático. De la misma forma, se describen una serie de sesiones prácticas donde los alumnos deberán utilizar *LabVIEW* como entorno de programación para resolver una serie de problemas planteados.

En la sección 2 del artículo se presenta una breve introducción a *LabVIEW* describiendo las ventajas que esta herramienta puede aportar a las tareas docentes. Posteriormente, las distintas sesiones prácticas y aplicaciones desarrolladas con *LabVIEW* son descritas en la sección 3. La sección 4 presenta un breve resumen de algunos resultados obtenidos por los estudiantes haciendo uso de las herramientas presentadas. Finalmente el artículo concluye con algunas conclusiones y futuros trabajos.

2 LABVIEW PARA TAREAS DOCENTES RELACIONADAS CON CONTROL AUTOMÁTICO.

Gracias al gran desarrollo y abaratamiento experimentado por los ordenadores personales y estaciones de trabajo en las últimas décadas, la forma de afrontar numerosos problemas de ingeniería y estudios científicos ha sufrido un cambio considerable. Se podría afirmar que un constante aceleramiento en la velocidad de cálculo e incremento en la capacidad de memoria disponible ha permitido que en las últimas décadas del siglo pasado y comienzos de éste, el foco haya sido progresivamente desplazado desde el hardware al software dando nombre a la hoy conocida como *Instrumentación Virtual*. La rápida adopción de los computadores en los últimos 20 años ha generado una importante revolución en la instrumentación de ensayos, mediciones y automatización. Un importante desarrollo resultante de la utilización del computador en conceptos de automatización, se conoce como *instrumentación virtual*, el cual ofrece variados beneficios a ingenieros y científicos que requieran mayor productividad, precisión y rendimiento [19].

Un instrumento virtual consiste en una computadora de tipo industrial, o una estación de trabajo, equipada con potentes programas (software), hardware económico y drivers, que cumplen en conjunto las funciones de instrumentos tradicionales. Los instrumentos virtuales representan un traslado fundamental de los sistemas de instrumentación basados en hardware a sistemas centrados en software que aprovecha potencia de cálculo, productividad, exhibición y capacidad de conexión a estaciones de trabajo

[19]. Aunque el computador ha experimentado avances significativos, es el software el que realmente provee la ventaja para construir, sobre esta potente base de hardware, instrumentos virtuales. Con los instrumentos virtuales, los ingenieros y científicos construyen sistemas de medida y automatización que se ajustan exactamente a sus necesidades (definidas por el usuario) en lugar de estar limitados por los instrumentos tradicionales (definidos por el fabricante).

Actualmente *LabVIEW* es el entorno de programación que refleja más fielmente este concepto de instrumentación virtual, donde sus propios programas se conocen como *VI* o *Virtual Instrument*. *LabVIEW* de *National Instruments* (NI) es un entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medida y presentación de datos. Además proporciona mucha mayor versatilidad que los lenguajes de propósito general (C o Java) con una mayor intuitividad y amigabilidad al no estar basado en líneas de código sino en un lenguaje gráfico, G, que abstrae al usuario de peculiaridades más propias de Ingeniería Informática, y habilita el desarrollo desde un punto de vista funcional, mediante diagramas de bloques y simple “Coger y Colocar” (*Pick and Place*) [11], [12].

Otra de las principales ventajas de *LabVIEW* es que permite abarcar toda la pirámide industrial, siendo posible realizar implementaciones de bajo nivel donde se trabaja directamente con los lazos de control, hasta la capa de supervisión y monitorización. De esta forma se presenta como una herramienta excelente para ser usada con fines docentes dentro del campo de la Automática, donde los estudiantes pueden poner en práctica sus conocimientos teóricos a distintos niveles de abstracción. A día de hoy es posible encontrar numerosas aplicaciones realizadas con *LabVIEW* con fines docentes, donde la gran mayoría de ellas se centran en laboratorios virtuales y remotos desarrollados de forma casi transparente gracias a las capacidades de *LabVIEW* para estos fines [3], [8], [17], [18]. La compañía NI es consciente de este hecho, y en los últimos años esta apostando fuertemente con la incorporación de nuevos paquetes dedicados exclusivamente a temas de simulación y control automático, aportando incluso conexiones entre *LabVIEW* y entornos ampliamente utilizados en esta área tales como *Matlab/Simulink*.

De esta manera, los principales beneficios de *LabVIEW* desde el punto de vista docente son los siguientes [11], [12]:

- Alta componente interactiva que permite realzar la motivación de los estudiantes.
- Proporciona funcionalidad interna de medida y análisis que facilita a los estudiantes un aprendizaje práctico.
- Provee conectividad abierta con otras herramientas de ingeniería ampliamente extendidas como *Matlab/Simulink* o *Excel*.
- Es altamente amigable y fácil de utilizar lo cual permite al docente concentrarse en la teoría y emplear menos tiempo en programación.
- Permite la creación de laboratorios remotos y virtuales de forma totalmente transparente.
- Teniendo en mente que uno de los principales objetivos de la enseñanza de alumnos de ingeniería y ciencia es el surtir a la industria de profesionales capacitados, *LabVIEW* es una herramienta cada vez más extendida y demandada.

3 HERRAMIENTAS DOCENTES DESARROLLADAS CON LABVIEW

Esta sección describe el conjunto de herramientas que han sido desarrolladas haciendo uso de *LabVIEW* y que se utilizan como soporte didáctico en distintos estudios de ingeniería en la Universidad de Almería.

3.1 MODELADO Y CONTROL CLMÁTICO DE UNA MAQUETA DE INVERNADERO

3.1.1 Justificación

Como ya es bien conocido, la agricultura, y más concretamente el cultivo bajo invernadero, es uno de los motores económicos principales de la provincia de Almería, algo que ha llevado a consolidar los estudios de agronomía como una de las referencias principales de la universidad almeriense. En la formación de un ingeniero técnico agrícola o ingeniero agrónomo es fundamental hoy día poseer conocimientos relacionados con modelado y control, debido a la presencia, cada día más frecuente, de sistemas de control automático en los invernaderos. Es por ello que desde el área de Ingeniería de Sistemas y Automática de la Universidad de Almería se esta apostando fuertemente en mejorar la docencia en este campo, trasladando los avances obtenidos en los campos de modelado y control gracias a las nuevas tecnologías [8], [13], [14].

En esta práctica se pretende describir los pasos necesarios para lleva a cabo el desarrollo de un sistema de control, su implementación y prueba en una instalación real. Concretamente, se utilizará como planta una maqueta de invernadero en la que

se han instalado sensores y sistemas de actuación para el control de las variables climáticas en su interior y de fertirrigación, con el fin de reflejar lo más fielmente posible el comportamiento climático de un invernadero real. Como controlador se utilizará un computador en el que se programarán los algoritmos de control diseñados. Con esta práctica el alumno desarrollará un diseño completo del sistema de control de un determinado proceso, desde la obtención del modelo realizando ensayos reales en una planta hasta su implementación y prueba, pasando por todas las fases intermedias en las que realizará ensayos de simulación del sistema para obtener los parámetros característicos del mismo.

El objetivo principal de la práctica se resume en el desarrollo de un proyecto de control completo de una variable climática en el interior de un invernadero o de fertirrigación de cultivos. Los problemas de control disponibles en la maqueta de invernadero son los siguientes [8]:

- Control de la temperatura diurna utilizando ventilación natural.
- Control de la temperatura nocturna utilizando un sistema de calefacción.
- Control de la radiación incidente en el cultivo utilizando una malla de sombreo.
- Control de riego bajo demanda.

Esta práctica se utiliza en tercer curso de Ingeniería Técnica Agrícola dentro del marco de la asignatura *Control y Robótica en Agricultura*, así como en un curso de doctorado internacional impartido para alumnos de Sudamérica denominado *Control Automático y Robótica en Agricultura*.

3.1.2 Descripción de la herramienta

Descripción de la maqueta de invernadero

La maqueta representa un invernadero multicapilla, de cubierta curva asimétrica a dos aguas con orientación Este-Oeste y de dimensiones 25 x 50 m (ver Figura 1). Concretamente, se trata de una estructura construida dentro del proyecto *DAMOCIA* (Diseño Asistido Mediante Ordenador para la Construcción de Invernaderos Automatizados) [13].

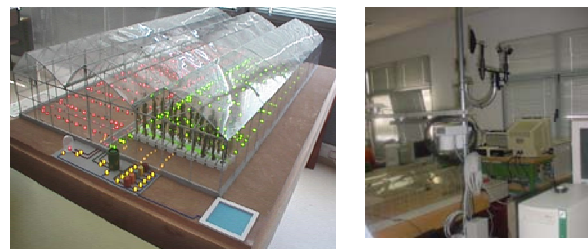


Figura 1. Maqueta de Invernadero.

La maqueta esta dotada de una serie de sensores y actuadores que permiten simular el clima de un invernadero real. De esta manera es posible tomar medidas de temperatura con una Pt100, de humedad con un higrotermotransmisor potenciométrico, y de radiación con sensores basados en termoelementos (radiación global y neta) y fotodiodos (radiación PAR). En cuanto a actuadores se refiere actualmente la maqueta de invernadero dispone de los siguientes: ventilación lateral, ventilación cenital, mallas de sombreo, calefacción en los dos sectores, instalaciones de riego y fertilización para dos sectores independientes [8].

Para aproximar la maqueta a un sistema real se han incorporado a la maqueta un foco de 500 W haciendo la función del sol, una resistencia de un secador de cabello para poder calentar el aire interior del invernadero y un ventilador de una fuente de alimentación haciendo la función de ventilación forzada. Los tres dispositivos son controlados con relés [8].

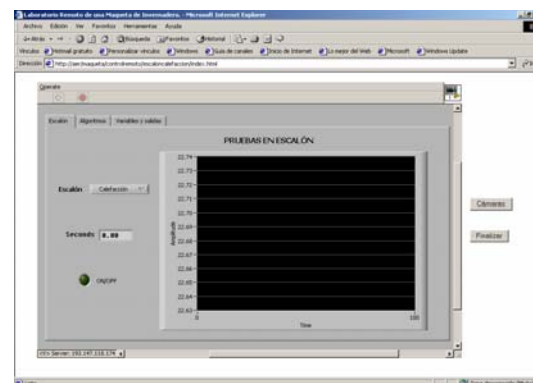
Herramienta desarrollada

Consiste en un laboratorio remoto desarrollado en *LabVIEW* que permite a los estudiantes llevar a cabo los principales objetivos de la práctica: ensayos sobre la variable climática a controlar e implementación del algoritmo de control correspondiente. De esta manera, la herramienta desarrollada consta de tres pantallas diferentes tal y como se observa en la Figura 2.

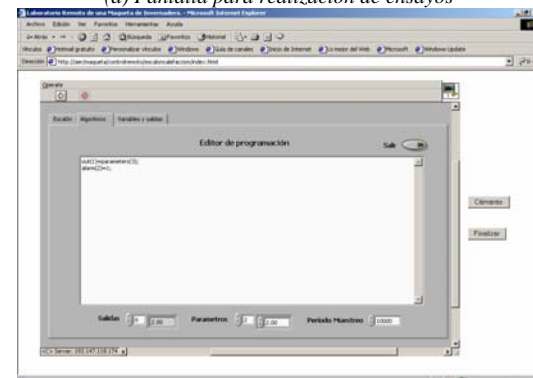
Para llevar a cabo un ensayo sobre la variable climática deseada, el alumno accederá al sitio web de la herramienta y deberá seleccionar el tipo de ensayo a realizar en el menú desplegable *Escalón* mostrado en la Figura 2(a), cuyas opciones son: calefacción, ventilación y radiación. Tras dicha selección el alumno realiza el ensayo correspondiente, descarga los datos en formato *ASCII* ordenados por columnas, y los analiza haciendo uso de *Matlab* atendiendo a la información mostrada en la Tabla 1.

Tabla 1. Relación de datos obtenidos en un ensayo sobre la Maqueta de Invernadero.

Columna	Sensor	Unidades
1	Temperatura exterior	°C
2	Humedad relativa interior	%
3	Temperatura interior	°C
4	Radiación global interior	W/m ²
5	Velocidad del viento	m/s
6	Lluvia	-
7	Temperatura tubos de calefacción	°C
8	Dirección del viento	°



(a) Pantalla para realización de ensayos



(b) Pantalla para implementación del algoritmo de control



(c) Pantalla para lectura de sensores

Figura 2. Laboratorio Remoto para Modelado y control de variables climáticas de invernadero.

Una vez analizados los datos obtenidos del ensayo seleccionado, los estudiantes implementarán el algoritmo de control pertinente haciendo uso del editor remoto mostrado en la Figura 2(b). Esta pantalla permite a los estudiantes implementar en código *Matlab* el algoritmo de control, utilizando una serie de variables preestablecidas para acceder a los sensores (variable *sensores*) y modificar el estado de los actuadores (variable *out*). De esta manera una vez implementado el algoritmo de control, el lazo se cierra haciendo uso de estas variables. Existen otra serie de variables utilizadas para indicar alarmas y algunas otras auxiliares disponibles para realizar cálculos extra, las cuales se pueden mostrar en la pantalla de visualización junto a los valores de las variables climáticas (ver Figura 2(c)). Una vez realizado el control, los

alumnos pueden descargar los datos en formato *ASCII*, de la misma forma que para los ensayos, y analizar los resultados haciendo uso de *Matlab*. Con el fin de aumentar la motivación del alumno, el laboratorio remoto esta dotado de una serie de cámaras que permiten obtener realimentación visual, tal y como se muestra en la Figura 3 [8].



Figura 3. Cámaras para maqueta de invernadero.

3.2 MODELADO Y CONTROL DE UN MOTOR DE CC

3.2.1 Justificación

Una de las principales vías que permite a los estudiantes comprender los contenidos matemáticos estudiados en clases teóricas se centra en poner en práctica dichos conocimientos. Para ello, esta práctica tiene como objetivo la aplicación de los conceptos fundamentales de modelado y análisis de sistemas dinámicos a un sistema real, concretamente a un motor de corriente continua (CC) que es un servomecanismo de posicionamiento angular. Las principales tareas se centran en obtener el modelado de la velocidad del motor con respecto a la entrada de tensión en la armadura del mismo, tanto de forma teórica como basada en los principios físicos de funcionamiento del mismo para comparar las respuestas reales con las que proporciona el modelo. Además, se realizará un análisis de la respuesta del sistema en el dominio del tiempo, así como en el dominio de la frecuencia. Una vez consolidado el modelado del motor, los estudiantes llevarán a cabo el control en velocidad y posición del mismo.

Esta práctica se imparte en segundo curso de Ingeniería Técnica en Informática de Sistemas y en quinto curso de Ingeniería Química, bajo el marco de las asignaturas *Control por Computador* [15] y *Control Avanzado de Procesos Químicos*.

3.2.2 Descripción de la herramienta

Descripción del motor de corriente continua

En el laboratorio de Automática, Robótica y Visión Artificial del Departamento de Lenguajes y Computación de la Universidad de Almería existen 12 puestos de trabajo dotados con motores de CC

(ver Figura 4) que permiten llevar a cabo la presente práctica. Estos motores pertenecen a la empresa *Kelvin*, donde las principales características técnicas desde el punto de vista práctico son las siguientes [15]:

- Referencia motor: 2332 24 V con dinamo-codificador 3604.
- Voltaje nominal: 24 V.
- Velocidad máxima sin carga: 5750 rpm.
- Constante de velocidad: 243 rpm/V.
- Tacodinamo con constante tacométrica: 0.52V/1000 rpm. Suministra una tensión proporcional a la velocidad angular de giro.
- Codificadores *HEDS 5540A 3C 500 L*. Dispone de un conector para medida de posición angular.
- Tarjeta de control servoamplificadora *CMC24,2* con ganancia y velocidad máxima regulable por potenciómetro y transformador de 1.5 A.

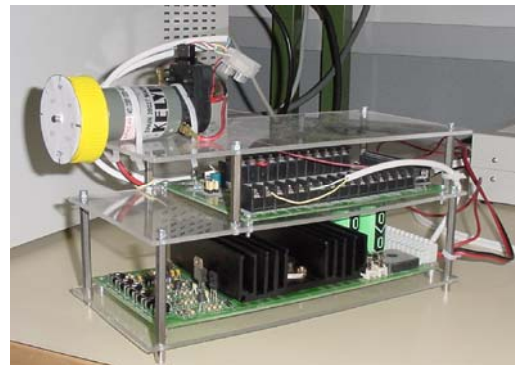


Figura 4. Motor de corriente continua

Herramienta desarrollada

Para llevar a cabo el análisis de la respuesta dinámica de los motores y el control de los mismos, se dispone de una herramienta implementada utilizando *LabVIEW*, de modo que se proporciona al alumno un entorno amigable con el que proceder a la realización de ensayos. La herramienta permite intercambiar los datos con *Matlab*, así como guardarlos en archivos de texto. Una imagen de la interfaz principal de la herramienta se muestra en la Figura 5, en la que se pueden apreciar los principales elementos de los que consta el programa. Lo primero que se observa es la sección correspondiente a las gráficas, que se encuentra en la parte superior, donde se representarán los resultados del ensayo [1].

En la parte inferior derecha se encuentran los dos botones principales de la aplicación, el de *PARAR* para finalizar y el de *INICIAR* para empezar la simulación. Encima de ellos se encuentra un control numérico que indica en segundos el tiempo de simulación deseado. En la parte inferior

izquierda hay un control llamado 'Dial', que indica la tensión que se le proporciona al motor cuando se está controlando el servomotor en bucle abierto.

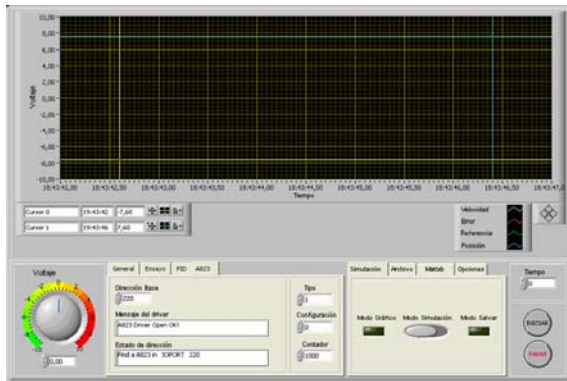


Figura 5. Aplicación para el modelado y control del motor de corriente continua.

Entre el dial para indicar la tensión y los botones para arrancar y parar el programa se encuentran dos conjuntos de fichas con las opciones de configuración de la aplicación y la definición de los experimentos a realizar, de las cuales las más significativas se comentan a continuación [1]:

- *Ensayo*. En esta ficha se indicará si es preciso que en el ensayo se introduzca una entrada del tipo impulso, una rampa, una senoidal o de uno a tres escalones.
- *PID*. Activa el PID y sirve para configurar los parámetros de éste (K_p , T_i y T_d).
- *A823PG/PCI-1202*. Ficha para la configuración de la tarjeta A/D.
- *Simulación*. Permite configurar la visualización de las señales en tiempo real durante el ensayo.
- *Archivo*. Permite guardar los resultados de un ensayo en un archivo o cargar unos resultados anteriores que previamente han sido guardados en un archivo de texto.
- *Matlab*. En esta ficha se indica si los resultados del ensayo se envían directamente a *Matlab*.

Por tanto, los alumnos en primer lugar utilizarán la herramienta para realizar los distintos ensayos en bucle abierto, exportarán los datos a *Matlab*, analizarán los resultados y obtendrán el modelo del sistema. Una vez obtenido el modelo, diseñarán un controlador PID atendiendo a una serie de especificaciones y utilizarán la herramienta para llevar a cabo el control en velocidad y posición del motor en CC. Los resultados de control se exportan de nuevo a *Matlab* para ser analizados y documentados.

3.3 DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA SCADA

3.3.1 Justificación

Las exigencias que actualmente se imponen a los procesos productivos en cuestión de rendimiento, calidad y flexibilidad, hacen necesario introducir las nuevas tecnologías en el control y vigilancia de éstos. Con este propósito, nace la idea de supervisar los procesos. La incorporación de nuevas tecnologías en la industria permite la reducción del número de paradas innecesarias, la predicción de citaciones anómalas o la actuación rápida y eficaz de forma que se asegure la continuidad y uniformidad de la producción. Así, la supervisión de procesos se establece como forma de automatizar tareas como las descritas en las guías de aseguramiento de la calidad y/o en los planes de mantenimiento preventivo con el fin de eliminar o reducir situaciones indeseadas.

La centralización y registro de datos es el primer paso en la implantación de un sistema de supervisión, y su simplicidad reside en la conectividad que ofrecen los actuales sistemas de control. Son los llamados sistemas *SCADA* (*Supervisory Control And Data Acquisition*) o software de monitorización y control que permiten el acceso a datos del proceso y cierta interacción entre el operario (interfaces gráficas y animadas) y el proceso (adquisición de datos a través de dispositivos de campo). Estos sistemas *SCADA* han sustituido las salas de control por ordenadores o terminales de control, y los bellos pero estáticos y voluminosos sinópticos por pantallas configurables y animadas. El objetivo es uno: facilitar la tarea del operario encargado de la vigilancia del proceso y su seguimiento.

El fin de esta práctica consiste en el análisis, diseño e implementación de una herramienta de supervisión y control utilizando *LabVIEW* e incorporando las principales características de un *SCADA*. Por tanto, los principales objetivos de esta práctica se resumen en:

- Identificación de las variables de un proceso industrial propuesto.
- Modelado y análisis de un sistema dinámico y el diseño de controladores del mismo.
- Diseño e implementación de una herramienta *SCADA* para la monitorización del mismo que incorpore las funciones básicas de este tipo de aplicaciones.

Esta práctica se imparte en cuarto curso de Ingeniería Informática, en una asignatura denominada *Informática y Automática Industrial*.

3.3.2 Descripción de la herramienta

Descripción de planta virtual

Toda herramienta SCADA esta encargada de monitorizar, controlar y supervisar uno o varios procesos de control. En esta práctica se ha propuesto una planta virtual, que los alumnos utilizarán para cumplimentar los objetivos establecidos. El sistema propuesto se muestra en la Figura 6.

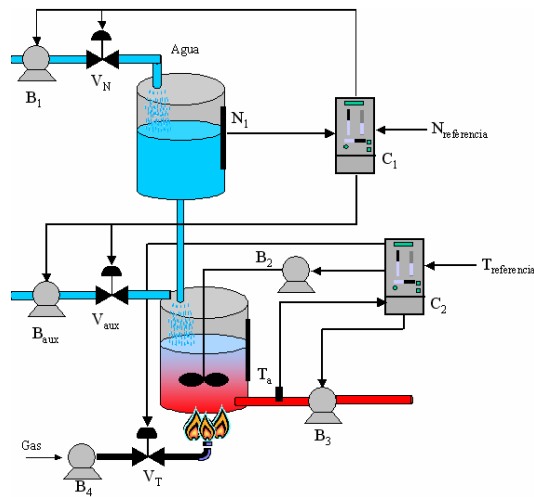


Figura 6. Planta virtual de tanques acoplados.

En este sistema se controla el nivel del tanque superior, N_1 , mediante la apertura de la válvula proporcional, V_N , con el fin de que el caudal de entrada del tanque inferior sea constante. El objetivo final del sistema es el control de la temperatura del agua de salida, T_a , de este último tanque mediante la apertura de la válvula proporcional V_T . En caso de que el nivel del tanque superior se encuentre por debajo de un determinado valor, debe activarse el circuito auxiliar encendiendo la bomba B_{aux} y la válvula todo/nada V_{aux} . Ya que la planta propuesta no se encuentra disponible en la Universidad de Almería, el comportamiento dinámico de la misma se ha simulado utilizando *Simulink* tal y como se observa en la Figura 7. Este archivo esta protegido de tal manera que los alumnos no pueden visualizar el código, donde la única forma de conocer la dinámica de planta es realizando ensayos sobre la misma como si de una planta real se tratase.

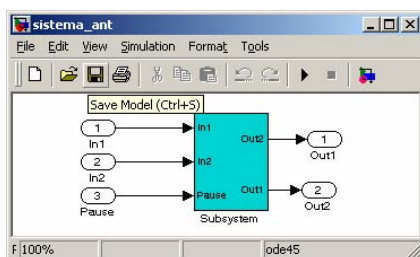
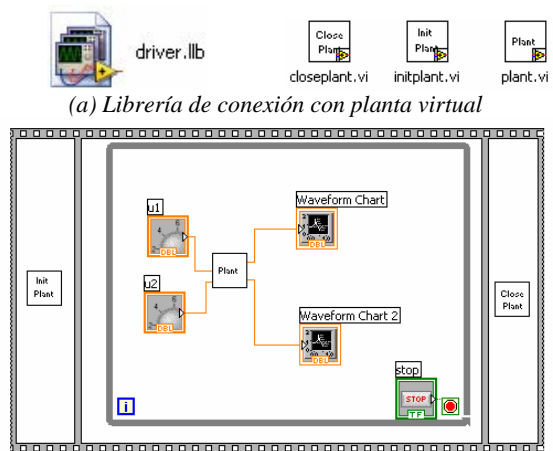


Figura 7. Planta virtual en Simulink

Herramienta desarrollada

Para permitir llevar a cabo las distintas etapas de la práctica, se ha desarrollado una librería en *LabVIEW* (*driver.llb*) que permite el acceso al modelo en *Simulink* comentado anteriormente. Esta librería contiene tres funciones o VIs (*Virtual Instruments*) que simulan las acciones básicas que se tendrían que realizar para conectarse con una planta real (ver Figura 8(a)): *initplant.vi*, para inicializar el acceso a la planta; *plant.vi*, para acceder a los actuadores de la planta y adquirir las variables del proceso; y *closeplant.vi*, para cerrar la conexión con el sistema. La Figura 8(b) muestra un ejemplo en código *LabVIEW* donde se hace uso de la librería desarrollada para acceder a la planta virtual y poder realizar ensayos en bucle abierto.



(b) Ejemplo de acceso a planta virtual con LabVIEW.

Figura 8. Librería en LabVIEW para simular la conexión con la planta virtual.

Por tanto, en primer lugar los alumnos dedican unas primeras sesiones de prácticas a aprender a utilizar *LabVIEW* como entorno de programación. Posteriormente llevan a cabo el modelado de la planta virtual haciendo uso de la librería comentada anteriormente, e implementan un par de controladores desacoplados para el control de nivel y temperatura del proceso. Finalmente, desarrollan un sinóptico del sistema, y dotan a la herramienta resultante de las características básicas de un SCADA: monitorización, control manual, gestión de alarmas, prioridades, perfil de usuario, almacenamiento de datos y su acceso, generación de informes, etc.

3.4 CONTROL MULTIVARIABLE DE UNA MAQUETA DE CUATRO TANQUES

3.4.1 Justificación

Uno de los campos de interés en la Ingeniería de Sistemas y Automática son las técnicas de control multivariable. Existen multitud de procesos industriales, especialmente en la industria

aeronáutica y química, cuya estabilidad depende del estado de varias variables y en donde la aplicación de técnicas de control clásicas puede producir resultados poco eficientes [4]. El proceso de los cuatro tanques desarrollado por Johansson ilustra de forma muy clara e intuitiva las ventajas y limitaciones de rendimiento que poseen las técnicas de control multivariable, y se presenta como una excelente herramienta para la docencia de este tipo de técnicas de control. Con este sistema es posible también visualizar la localización y dirección de los ceros, y su repercusión sobre la estabilidad del sistema.

Esta práctica esta dedicada al uso del proceso de la maqueta de cuatro tanques para comprender el modelado y control de sistemas multivariables, centrando el interés en:

- Conocer el *RGA* (*Relative Gain Array*).
- Control por desacoplo y control desacoplado.
- Limitaciones impuestas por sistemas de fase no mínima.

Los contenidos de esta práctica se imparten en la asignatura *Control Avanzado de Procesos Químicos* de la carrera de quinto de Ingeniería Química, así como en el curso de doctorado *Técnicas avanzadas en Automática, comunicaciones para control y robótica*.

3.4.2 Descripción de la herramienta

Descripción maqueta de cuatro tanques

En los últimos años, el Instituto de Automática y Fabricación de la Universidad de León, en colaboración con la UNED y la Universidad de Almería han desarrollado una maqueta del proceso de los cuatro tanques con el objetivo de su utilización con fines docentes y de investigación, la cual se puede observar en la Figura 9. Esta planta constituye un excelente campo de ensayo de estrategias de control multivariable a nivel industrial.

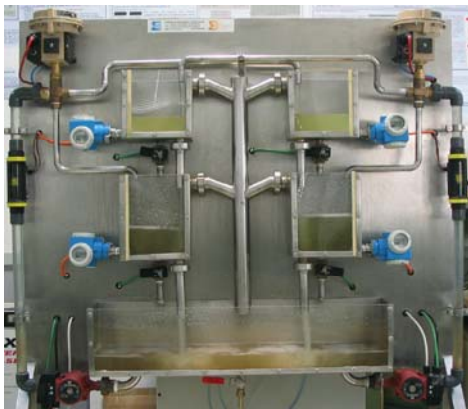


Figura 9. Maqueta de cuatro tanques.

La maqueta intenta mantener la estructura original a la propuesta por Johansson [10], donde los principales elementos de la misma se resumen en lo siguiente:

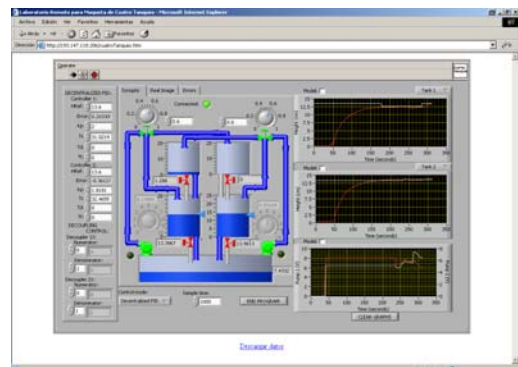
- Bombas *Grundfos* UPE 25-40 con módulos de expansión MC 40/60, regulables mediante una señal analógica.
- Válvulas neumáticas de tres vías *Samson* 3226.
- Transmisores de presión *Endress & Hauser* PCM 731 como sensores de nivel.
- Electroválvulas todo/nada *SMC*, instaladas en la base de los cuatro tanques principales para permitir la aplicación de perturbaciones.
- Sistema de adquisición *Snap Ultimate I/O* de *Opto 22*, que permite la conexión con cualquier topología de red física así como integrarse en todo tipo de redes empresariales o industriales.

Herramienta desarrollada

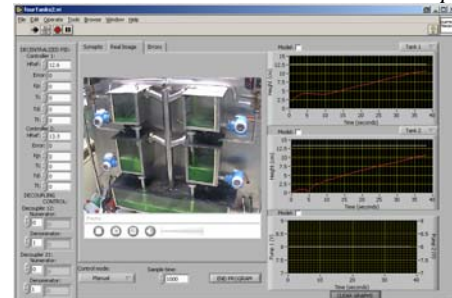
Se ha desarrollado una herramienta *SCADA* para la monitorización y supervisión de la planta utilizando *LabVIEW*, con el fin de permitir el llevar a cabo los principales aspectos prácticos que se comentaron anteriormente [4]. En la Figura 10 se muestran varios aspectos de la aplicación desarrollada.

Los aspectos más importantes de la herramienta desarrollada son:

Adquisición y registro de datos: cada período de muestreo, se registran remotamente las señales de los sensores y actuadores.



(a) Acceso al laboratorio vía web. Vista del sinóptico.



(b) Realimentación visual de la maqueta.

Figura 10. Laboratorio Remoto para el control de la planta de cuatro Tanques.

Representación del proceso: los datos adquiridos del sistema real son presentados gráficamente en la aplicación, con la finalidad de mostrar su valor actual, su evolución y facilitar al usuario su interpretación. De la misma manera existe la posibilidad de obtener realimentación visual del proceso, ya que la aplicación ofrece en tiempo real la imagen de la maqueta tomada a través de una cámara AXIS 2111 instalada en el laboratorio (ver Figura 10(b)).

Interacción con el proceso: durante la ejecución del programa, el usuario puede interactuar con las bombas, válvulas de tres vías y válvulas todo/nada actuando directamente sobre la representación de estos elementos en el sinóptico de la Figura 10(a).

Control Multivariable: se han implantado las siguientes técnicas de control multivariable: PID desacoplados y control por desacoplo. Se selecciona el tipo de control mediante el menú desplegable “Control Mode” y, mientras esté llevándose a cabo el control, no se permite al usuario interactuar con las bombas.

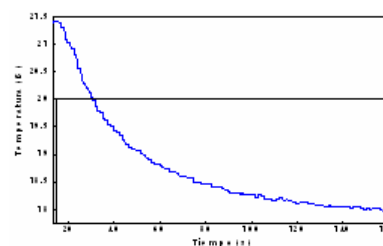
Por tanto, los alumnos pueden utilizar el laboratorio remoto desarrollado para realizar múltiples ensayos en bucle abierto de la planta, obtener el modelo lineal y no lineal de la misma, así como llevar a cabo el estudio de distintas técnicas de control multivariable.

4 RESULTADOS OBTENIDOS POR LOS ALUMNOS

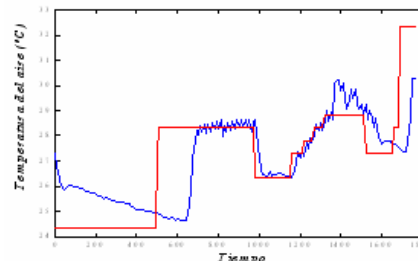
Las herramientas y sesiones prácticas descritas en la sección anterior, se han utilizado e impartido en la Universidad de Almería durante los últimos 4 años, obteniendo resultados satisfactorios por parte de los alumnos. En esta sección se describen algunos de los resultados prácticos realizados por los alumnos de las diferentes asignaturas.

4.1 MODELADO Y CONTROL DE TEMPERATURA DE LA MAQUETA DE INVERNADERO

En esta primera práctica, los alumnos utilizaron el laboratorio remoto mostrado en la Figura 2 con el fin de modelar y controlar la temperatura de la maqueta de invernadero haciendo uso del sistema de calefacción. De esta manera, en primer lugar realizaron un ensayo en bucle abierto del sistema aplicando un escalón de 0-100% obteniendo los resultados mostrados en la Figura 11(a). Haciendo uso de esta curva, calcularon el modelo del sistema, diseñaron un controlador PID, e implementaron dicho controlador en código *Matlab* haciendo uso del editor remoto (ver Figura 2(b)). El resultado de control obtenido se puede observar en la Figura 11(b).



(a) Modelado de temperatura con calefacción.



(b) Control de temperatura con calefacción.

Figura 11. Resultados de modelado y control de temperatura con calefacción.

4.2 MODELADO Y CONTROL DE UN MOTOR DE CC

De la misma forma que en la práctica anterior, en este caso los estudiantes realizan múltiples ensayos en bucle abierto sobre el sistema, pero ahora haciendo uso del motor de corriente continua, para obtener el modelo del mismo, y para posteriormente llevar a cabo el control de la planta. La Figura 12 muestra una batería de distintos ensayos a los que los estudiantes suelen someter al motor de corriente continua con el fin de obtener el modelo en velocidad y en posición del mismo, así como detectar las posibles no linealidades del sistema (saturación y zona muerta).

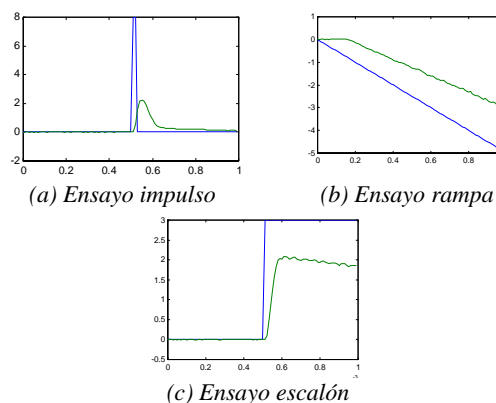
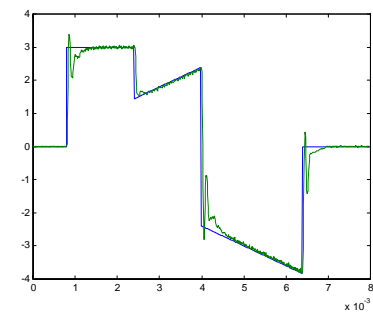
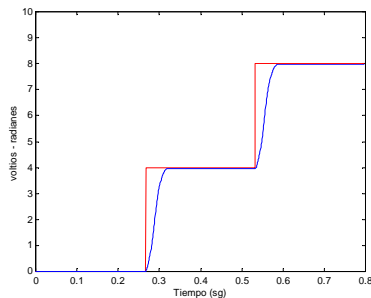


Figura 12. Respuestas de la velocidad del motor ante diferentes entradas

Una vez obtenido el modelo, y teniendo en cuenta el carácter no lineal del sistema, los estudiantes diseñan distintos controladores para velocidad y posición, obteniendo resultados como los que se muestran en la Figura 13.



(a) Resultado de control de velocidad.



(b) Resultado de control de posición.

Figura 13. Resultado de control de velocidad y posición del motor de CC.

4.3 DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA SCADA

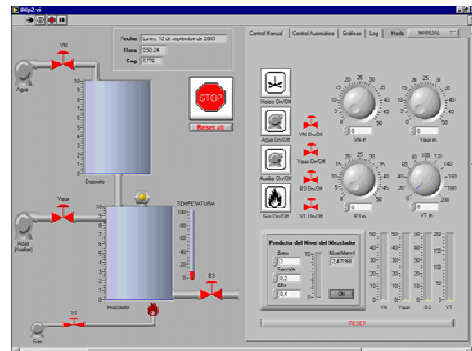
Como se comentó anteriormente, el objetivo final de esta práctica consistía en el desarrollo de una herramienta *SCADA* que permitiese el control y supervisión de la planta virtual propuesta. En la Figura 14 se muestra la pantalla principal y el código *LabVIEW* de una herramienta desarrollada por alumnos haciendo uso de la librería y la planta virtual descritas en la sección 3.3.

4.4 CONTROL MULTIVARIABLE DE UNA MAQUETA DE CUATRO TANQUES

Como paso previo al control multivariable de la planta de los cuatro tanques, los alumnos realizan una serie de ensayos haciendo de uso del laboratorio remoto presentado en la Figura 10, con el fin de obtener el modelo de la planta. En la Figura 15 se muestra el resultado del modelado del nivel de los tanques inferiores de la maqueta, donde se observa cómo el modelo se aproxima fielmente a los resultados reales.

Una vez obtenido el modelo de la planta, los alumnos diseñan y estudian en primer lugar el control de la planta con PID desacoplados, observando cómo el sistema sigue las referencias deseadas, pero donde el efecto de la interrelación entre variables es bastante considerable. Un ejemplo se muestra en la Figura 16(a).

Finalmente los alumnos hacen uso de la herramienta para implementar un control por desacople del sistema, comparando los resultados obtenidos con los PID desacoplados. Un ejemplo de ello se observa en la Figura 16(b) donde se puede ver como el efecto de una variable sobre otra ha sido considerablemente reducido.



(a) Pantalla de explotación de la herramienta SCADA.

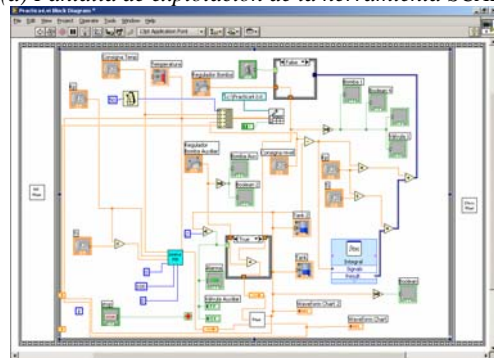
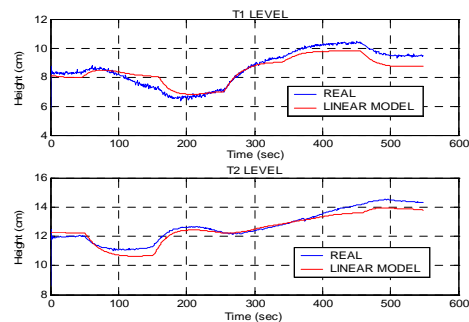
(b) Código LabVIEW de la herramienta SCADA.
Figura 14. Ejemplo herramienta SCADA.

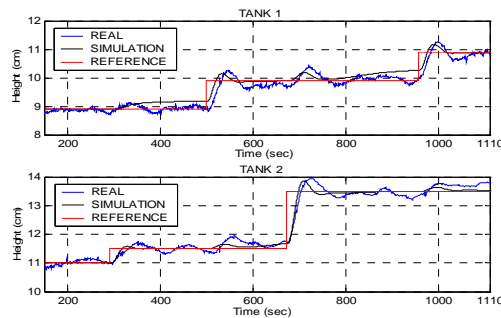
Figura 15. Comparación de proceso real con el modelo obtenido por los alumnos.

5 CONCLUSIONES

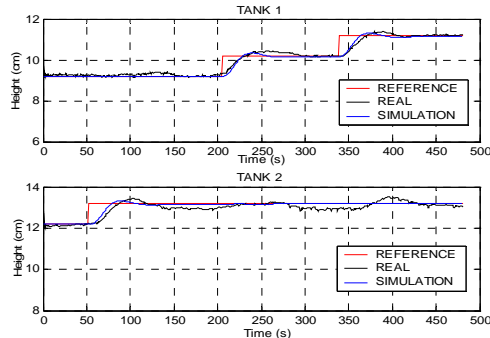
En este trabajo se ha presentado un conjunto de herramientas y sesiones prácticas desarrolladas haciendo uso de *LabVIEW*. Las distintas herramientas permiten abarcar múltiples aspectos relacionados con el campo del control automático, desde aquellos relacionados con aspectos básicos de modelado y control PID, hasta desarrollo de

herramientas SCADA o estrategias de control más avanzadas, como es el caso del control multivariable. Algunos de los trabajos futuros que se pretenden realizar para completar este conjunto de herramientas son:

- Laboratorio remoto para el control de una célula robotizada.
- Laboratorio virtual y remoto para introducir conceptos de tiempo real.



(a) Control con PID descaoplados



(b) Control por desacoplo.

Figura 16. Control por desacoplo de la maqueta de cuatro tanques.

Agradecimientos

Este trabajo ha sido financiado por los proyectos CICYT-FEDER DPI2002-04375-C03 y DPI2004-07444-C04-04.

Referencias

- [1] Álvarez, J.D. (2003). "Diseño e implementación de sistemas de control para procesos con dinámica no-lineal". Proyecto fin de Carrera de Ingeniería Informática, Universidad de Almería.
- [2] Dormido, S. (2004). "Control Learning: Present and Future". *Annual Reviews in Control*, Wiley, **28**(1), pp. 115-136.
- [3] Ertugrul, N. (2000) "Towards Virtual Laboratories: a Survey of LabVIEW-based Teaching-Learning Tools and Future Trends". *Int. J. Engng Ed.* **16**(3), pp. 171-180.
- [4] García, A.A.; Berenguel, M.; Guzmán, J.L.; Dormido, S.; Domínguez, M. (2006). "Remote laboratory for teaching multivariable control techniques". *Proceedings of the 7th IFAC Symposium on Advances in Control Education*, Madrid (España).
- [5] Guzmán, J.L. (2006). "Interactive Control System Design". Tesis doctoral. Universidad de Almería.
- [6] Guzmán, J.L.; Åström, K.J.; Dormido, S.; Hägglund, T.; Piguet, Y. (2006) "Interactive learning modules for PID control". *Proceedings of the 7th IFAC Symposium on Advances in Control Education*, Madrid (España).
- [7] Guzmán, J.L.; Berenguel, M.; Dormido, S. (2005). "Interactive teaching of constrained generalized predictive control", *IEEE Control Systems Magazine*, **25**(2), pp. 52-66.
- [8] Guzmán, J.L.; Berenguel, M.; Rodríguez, F.; Dormido, S. (2005) "Remote Control Laboratory using a Greenhouse Scale Model". *Computer applications in engineering education*, Wiley, **13**(2), pp. 111-124.
- [9] Guzmán, J.L.; Rodríguez, F.; Berenguel, M.; Dormido, S. (2005). "Virtual Lab for Teaching Greenhouse climate Control". *16th IFAC World Congress*. Prague. Czech Republic.
- [10] Johansson, K.H. (2000). "The quadruple tank process: A multivariable laboratory process with an adjustable zero". *IEEE Trans. on Control Systems Tech.*, **8**(3), pp. 456-465.
- [11] LabVIEW, National Instruments. <http://www.ni.com/labview/>.
- [12] LabVIEW Introduction Course. National Instruments. http://www.ni.com/academic/resources_lecture_materials.htm.
- [13] Rodríguez, F.; Berenguel, M.; Guzmán, J.L.; Dormido, S. (2006) "A virtual course on automation of agricultural systems", *Int. J. Engng Ed. Special issue on agricultural-biosystem-biological engineering education*. Para aparecer en 2006.
- [14] Rodríguez, F.; Berenguel, M. (2004). "Control y Robótica en Agricultura". Edt: Servicio de Publicaciones de la Universidad de Almería.
- [15] Rodríguez, F.; Berenguel, M.; Guzmán, J.L.; Arahal, M.R. (2003). "Prácticas de Control por Computador". Edt: Servicio de Publicaciones de la Universidad de Almería.
- [16] Sánchez, J.; Morilla, F.; Dormido, S.; Aranda, J.; Ruipérez, P. (2002). "Virtual Control Lab Using Java and Matlab: A Qualitative Approach", *IEEE Control Systems Magazine*, vol. 22, n° 2, pp 8-20.
- [17] Schwartz, T.L.; Dunkin, B.M. (2000). "Facilitating Interdisciplinary Hands-on Learning using LabVIEW". *Int. J. Engng Ed.* **16**(3), pp. 218-227.
- [18] Tan, K.K.; Lee, T.H.; Leu, F.M. (2000) "Development of a Distant Laboratory using LabVIEW". *Int. J. Engng Ed.* **16**(3), pp. 273-282.
- [19] Tracnova, S.A. (2003) "La Instrumentación Virtual". National Instruments.

LAS TECNOLOGÍAS XML EN LA AUTOMATIZACIÓN INDUSTRIAL

Elisabet Estévez

Marga Marcos, Darío Orive, Isabel Sarachaga

Escuela Técnica Superior de Ingeniería de Bilbao (Universidad del País Vasco)

Alameda Urquijo s/n 48013 Bilbao

Teléfono: 34 94 601 4049; Fax: 34 94 601 4187

e-mail: marga.marcos@ehu.es

Resumen

En este trabajo se analiza el uso de las tecnologías XML en aplicaciones de control industrial. Se ilustra el uso de las mismas en la creación de una infraestructura sobre la que desarrollar un entorno de soporte al desarrollo de este tipo de sistemas. El núcleo se basa en la definición de modelos de dominio que están relacionados describiendo conjuntamente un sistema de control industrial (SCI). Las herramientas se integran a través de la generación o consumo de parte de la información contenida en el modelo, la que corresponde a su dominio. Se presentan las tecnologías XML involucradas en la definición de los lenguajes de descripción, la comprobación de la corrección de los modelos y las técnicas de integración de herramientas.

Palabras Clave: Controladores Lógicos Programables, Descripción de sistemas de control industrial distribuido, Ciclo de desarrollo.

1 INTRODUCCIÓN

El avance tecnológico en el campo de la microelectrónica, las tecnologías software y la tecnología de las comunicaciones sigue mejorando a un ritmo vertiginoso debido a la demanda creciente de mercados masivos. Aunque con un cierto desfase, las tecnologías de la automatización han ido adoptando estos avances tecnológicos. Así, los sistemas de comunicación han ido reemplazando a gran parte del cableado, algunos dispositivos mecánicos han sido sustituidos por dispositivos electrónicos y muchos componentes software hacen las funciones que antes hacían componentes hardware. Esta adopción de tecnologías novedosas y probadas ha permitido además abordar aplicaciones de mayor complejidad.

Hoy en día, muchos sectores industriales utilizan los PLCs (Programmable Logic Controllers) para realizar el control de sus procesos productivos. En los últimos años los avances tecnológicos anteriormente comentados han aumentado espectacularmente las prestaciones de estos dispositivos de control, lo que ha redundado en la mejora y optimización del proceso productivo. Como consecuencia, se han reducido los costes de desarrollo y mantenimiento.

Ahora bien, debido a que el desarrollo de las aplicaciones de automatización sigue estando enfocado al dispositivo, no siempre es posible utilizar la potencialidad inherente a dichas tecnologías. Es más, a veces incluso impide su utilización. Por tanto, es claro que existe un vacío entre el estado del desarrollo tecnológico y las metodologías existentes para abordar el diseño y desarrollo de las aplicaciones. Como consecuencia, se hace necesario consolidar metodologías de diseño para los sistemas que controlan los procesos productivos. El establecimiento de metodologías permitirá adoptar los avances tecnológicos en microelectrónica, comunicaciones y software, reduciendo el tiempo de desarrollo y de salida al mercado de nuevos productos a un coste competitivo.

Paralelamente, han existido esfuerzos internacionales enfocados a la estandarización del desarrollo de aplicaciones basadas en PLCs. A principios de los 90 la IEC (International Electro-technical Commission) publicó diferentes partes del estándar IEC 61131 [16] que cubren el ciclo de vida de este tipo de aplicaciones. Consta de 5 partes: vista general, hardware, lenguaje de programación, guías de usuario y comunicaciones.

Concretamente, la parte 3 (publicada en 1993) está dedicada al entorno de desarrollo de software: la arquitectura software de la aplicación, los lenguajes de programación y la ejecución de los programas.

Con el fin de apoyar e impulsar el uso del estándar internacional en la programación de los autómatas

programables, se funda en 1992 la organización PLCopen (<http://www.plcopen.org>), en la actualidad constituida por más de 100 miembros, entre los que se encuentran fabricantes de PLCs, compañías dedicadas a la producción de software y otras instituciones independientes.

Desde la aparición del estándar IEC 61131-3 se ha observado una evolución de los sistemas de programación hacia la conformidad con el estándar. Pero en la actualidad aún existen sistemas de control cuya programación, tanto en cuanto a modelo software como a lenguajes de programación, es propietaria. Y, previsiblemente, será un largo proceso ya que tanto las aplicaciones existentes como los propios usuarios finales, que están acostumbrados a manejar entornos propietarios, están retrasando dicha implantación. Como consecuencia, resulta difícil integrar en una misma aplicación distintos sistemas de control. Lógicamente, es el usuario final el que sufre las consecuencias, entre las que destacan mayores costos de desarrollo y mantenimiento, escasa flexibilidad y falta de normalización en las soluciones de problemáticas de control industrial.

La mejora del proceso de diseño y desarrollo de aplicaciones de automatización complejas ha sido objeto de investigación de otros autores. Entre las diferentes aproximaciones publicadas para abordar el diseño de aplicaciones de automatización complejas, caben destacar las que utilizan UML [3] para modelar componentes IEC del sistema de control [9], [2], [20], [21], [22]. Heverhagen y Tracht [9] y Bonfé y Fantuzzi [2] utilizan UML para especificar componentes de sistemas de control conformes al estándar IEC 61131-3. Gonzalez y colaboradores [8] proponen una metodología para el análisis y modelado de sistemas de eventos discretos. En Kandare [12] se describe una herramienta gráfica que permite generar código en el lenguaje Texto Estructurado.

Por otro lado, se está trabajando en el diseño y desarrollo de entornos de soporte del proceso de desarrollo de sistemas de control industrial distribuido [20], [21], [22]. El entorno es orientado a objetos y utiliza UML para definir una arquitectura de cuatro niveles para diseñar los sistemas de interés.

Uno de los mayores inconvenientes de uso de este lenguaje de modelado, viene derivado de su potencialidad. La gran variabilidad de diagramas y elementos que permiten modelar cualquier tipo de aplicación y situación hace que sea demasiado complejo para ser utilizado por usuarios no expertos, como por ejemplo los técnicos que desarrollan aplicaciones industriales.

Trabajos previos de los autores también plantean el uso de UML para la definición de modelos de la aplicación desde diferentes dominios: funcional, hardware y software [15]. Con objeto de disminuir la complejidad del uso de herramientas UML, se propuso además una metodología, acotándose los diagramas a utilizar. La caracterización de los elementos UML utilizados se agrupa en Perfiles UML que añaden las características posibles a las particularidades de las aplicaciones a modelar.

En este trabajo se presenta una aproximación que también se basa en la definición de modelos pero, en este caso, el propio modelo de la aplicación es el medio para integrar las herramientas que intervienen en el diseño. La descripción interna de los modelos es independiente de las herramientas integradas, y la infraestructura ofrece mecanismos para definir del modelo de la aplicación, comprobar su corrección (dado que los modelos de dominio que lo forman pueden provenir de diferentes herramientas) y mecanismos para extraer/generar parte del modelo, que en última instancia, son los que permiten realizar la integración de herramientas. Y todo ello, de forma transparente al usuario.

Todas estas condiciones las cumplen las tecnologías XML [1],[10],[4],[5],[18],[19],[23],[24], añadiendo además, entre otras ventajas, la posibilidad de visualización y edición gráfica (completando la capa de abstracción con un interfaz gráfico y fácil de usar), la facilidad de mantenimiento y extensibilidad del entorno para adaptarlo a cambios de herramientas o a la evolución de las mismas, y utilizando estándares y software libre.

En el apartado 2 se analizan las características principales que definen la arquitectura de los sistemas de control industrial distribuido tanto desde el punto de vista software como hardware. Posteriormente se describe la tecnología XML *schema* y se utiliza para definir el lenguaje utilizando el modelo de los Sistemas de Control Industrial (SCI). El apartado 3 se dedica a ilustrar como se puede comprobar la corrección de los modelos mediante la tecnología XML *schematron*. Por último, en el apartado 4 se describen las técnicas de integración de herramientas.

2 DEFINICIÓN DE MODELOS DE SISTEMAS DE CONTROL INDUSTRIAL (SCI)

En trabajos previos de los autores [6], [7], [14] se han identificado tres dominios bien diferenciados involucrados en el diseño de los SCI: el dominio de la ingeniería de control, que estudia las estrategias de control. El dominio de la ingeniería de software,

cuyo objetivo es definir el código que implementa el diseño funcional. Por último, el dominio de la ingeniería eléctrica-electrónica, encargado de la selección de la plataforma hardware, donde se ejecutará el código de la aplicación.

Este trabajo se centra en el modelo de la implementación: arquitecturas hardware y software. En los siguientes sub-apartados se presentan sus características principales, así como las relaciones entre los elementos que las componen. Por último, antes de definir el lenguaje que deben seguir los modelos, se comenta brevemente la tecnología seleccionada: XML *schema*.

2.1 Arquitectura Software

El estándar IEC 61131-3 agrupa elementos y lenguajes que permiten programar la funcionalidad del sistema de control independientemente de la plataforma en la que se ejecuta [13]. Este estándar, permite diseñar aplicaciones de control de forma jerárquica utilizando los elementos del modelo software que proporciona: Configuración, Recurso, Tarea, Unidad de Organización del Programa (*Program Organization Unit-POU*) y variables [11].

Una *Configuración* contiene la arquitectura software completa que corresponde al PLC que representa. La *Configuración* contiene al menos un *Recurso*. Cada *Recurso* contiene el código ejecutable estructurado en *Programas*. La ejecución temporizada se consigue asociando *Programas* y *Bloques Funcionales* (Function Block – FB) a *Tareas*. Todos los programas y FBs asociados a una misma tarea se ejecutan con el mismo periodo y prioridad. La Figura 1 presenta de forma genérica y esquemática la relación existente entre los distintos tipos de elementos de la arquitectura software.

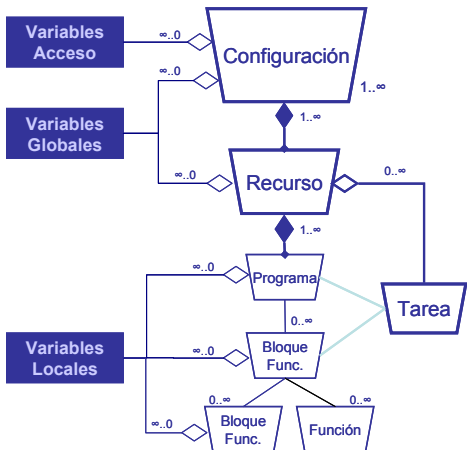


Figura 1: Elementos de modelado del software

Para concluir con la caracterización de la arquitectura software, en la siguiente tabla se presentan las características semánticas que se pueden (si son

opcionales) o deben (si son obligatorias) añadir en función del tipo de elemento.

Tabla 1: Características de los elementos software

Elemento SW	Características	
Configuración		
Recurso	Procesador donde se ejecuta (<i>OnProcessor</i>)	Opc.
Tarea	Periodo / Trigger	Obl.
	Prioridad	Obl.
Inst. POU	Tipo de POU	Obl.
	Tarea que lo organiza	Opc.
Variables	Tipo	Obl.
	Dirección física (<i>AT</i>)	Opc.
	Valor inicial	Opc.

2.2 Arquitectura Hardware:

La arquitectura hardware define los componentes que se deben utilizar para implementar la aplicación y la forma en que se deben conectar. Si el sistema es distribuido, estará compuesto por los distintos nodos y los diferentes segmentos de red que los comunican. Además, los componentes nodo estarán formados a su vez por un conjunto de componentes básicos. Estos elementos constituyen la parte de la aplicación dependiente del fabricante. Aunque todos ellos tienen características funcionales comunes, cada fabricante ofrece soluciones propietarias con diferentes posibilidades en cuanto a configuración y operación.

Entre los nodos se encuentran los llamados nodos de procesamiento, es decir, aquellos que ejecutan los algoritmos de control, y los nodos que recogen las entradas y salidas del sistema de control, es decir, nodos de entrada / salida. En general, los dos tipos de nodos están compuestos por los mismos elementos: fuentes de alimentación, procesadores, tarjetas de entrada / salida, tarjetas de memoria, tarjetas de comunicación, etc. La Figura 2 ilustra la estructura básica de cualquier aplicación industrial, identificando de forma general los elementos que forman un nodo. Así mismo, se observa que la tarjeta de comunicación es el elemento que une el nodo a un determinado segmento de bus.

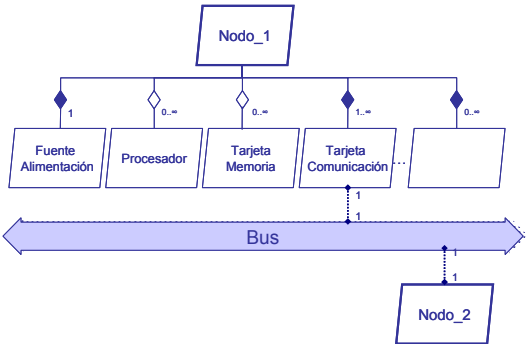


Figura 2: Arquitectura hardware

A partir de unas plantillas que contienen las características comunes a los elementos básicos, se pueden construir las plantillas correspondientes al equipamiento de un determinado fabricante añadiendo sus características particulares. En la siguiente tabla se presentan las características obligatorias e independientes de fabricante de los elementos que componen la arquitectura hardware.

Tabla 2: Características de los elementos Hardware

Elemento HW	Características	
Nodo	Referencia	Obl.
Bus	Protocolo Aplicación	Obl.
	Velocidad (Mps)	Obl.
Tarjeta Com.	Tipo (Profibus,...)	Obl.
	Dirección	Obl.
Otras Tarjetas	Referencia	Obl.

2.3 Relaciones entre las vistas de dominio

Una vez definidos los distintos elementos que participan en la definición de cada una de las vistas de dominio es necesario establecer las relaciones entre los diferentes elementos de cada una de ellas.

La Tabla 3 ilustra las relaciones más relevantes que se pueden dar entre las dos arquitecturas.

Tabla 3: Relaciones entre elementos HW y SW

Software	Hardware
Configuración	PLC
Recurso	Procesador
Variables globales con dirección física.	E/S

2.3 Conceptos básicos de la tecnología XML schema

XML es un lenguaje formal [1], [10], [24] y nació como un subconjunto de SGML (Standard Generalized Markup Language). Los datos contenidos en los documentos XML se organizan siguiendo una estructura jerárquica en forma de árbol. Las unidades mínimas de información que componen este árbol lógico son los *elementos*. Siempre existe un único elemento raíz que agrupa elementos hijos. Además, la información recogida en un elemento puede verse complementada a través de *atributos* (propiedades del elemento). Los atributos pueden ser de diferentes tipos, dependiendo de la naturaleza de la información que vayan a expresar.

XML permite definir lenguajes de marcado específicos para un uso determinado. Los requisitos gramaticales adicionales que deberá cumplir el documento XML escrito en el nuevo lenguaje se expresan en un *Schema* [23]. Un *Schema* recoge todas las restricciones léxicas y sintácticas que

definen el nuevo lenguaje. Por lo tanto, un documento XML será válido si, además de estar bien formado, respeta la estructura y restricciones que le imponga su *schema* asociado. Será un *parser* XML el encargado de comprobar la validez del fichero XML con respecto a un determinado *schema*.

En la siguiente figura se presenta a modo de ejemplo la definición de las características de las variables IEC 61131-3.

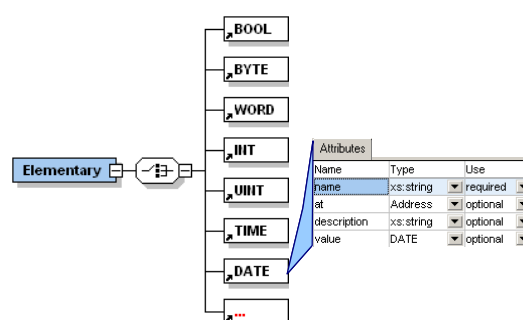


Figura 3: Características de las variables

Como se puede observar en la Figura 3, las variables se definen a través de un conjunto de atributos, concretamente uno por cada característica identificada en la Tabla 1. La tecnología *XML schema* puede ser utilizada para definir nuevos tipos de datos caracterizándolos por su formato de presentación combinado con un rango de valores. La Figura 4 ilustra la definición del rango de valores y posibles formatos del tipo *TIME* del estándar IEC 61131-3.

```
<xs:simpleType name="TIME">
  <xs:restriction base="xs:string">
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"d")"/>
    ..
    <xs:pattern
      value="(T|t)(TIME|time)(#d"d"d"p(P)?_)?d"m"d"p(P)?_)?d"s"d"p(P)?_)?d"ms")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"d"d"p(P)?_)?d"s"d"p(P)?_)?d"ms")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"h")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"hd"p(P)?_)?d"m(s|ms))"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"hd"p(P)?_)?d"m(d"p(P)?_)?d"(s|ms))"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"hd"p(P)?_)?d"m"d"p(P)?_)?d"ms")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"m")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"s")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"ms")"/>
    <xs:pattern value="(T|t)(TIME|time)(#d"p(P)?d"ms")"/>
  </xs:restriction>
</xs:simpleType>
```

Figura 4: Tipo de dato TIME

2.4 Infraestructura para modelar las arquitecturas software y hardware

La Figura 5 presenta una vista general de lenguaje de marcado definido con la tecnología *XML schema* (*SCI.xsd*), que marca las reglas sintácticas y de formato que debe seguir la descripción de un sistema de control industrial distribuido. La descripción global aúna las dos arquitecturas así como las posibles relaciones entre componentes de ambas.

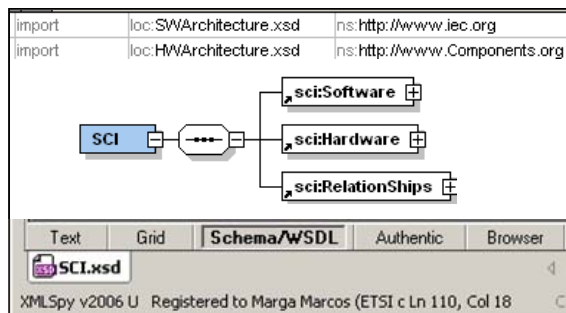


Figura 5: vista general de SCI.xsd

En cuanto a la arquitectura software y con objeto de hacer un modelado compatible con el propuesto por TC 6 XML [17], se ha separado la definición de la arquitectura software en dos elementos, tal y como se puede apreciar en la siguiente figura:

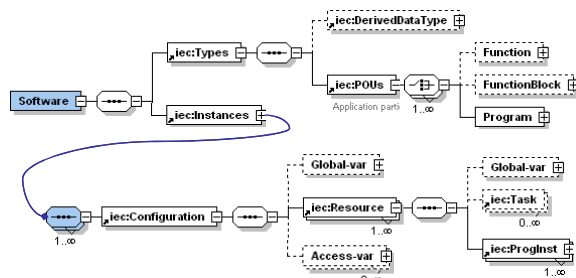


Figura 6: Arquitectura Software

Por un lado se define la parte dependiente del programador (*iec:Types* de la Figura 6) que contiene las variables y los POUs definidos por el programador. La Figura 7 se presenta la caracterización en XML de los POUs tipo bloque funcional y programa.

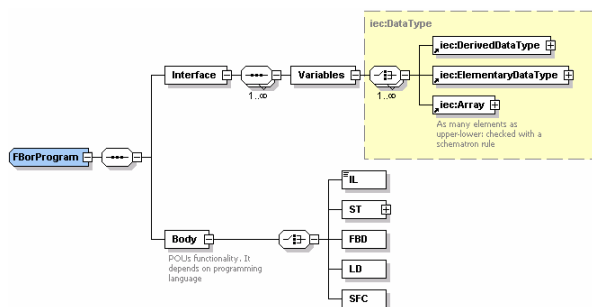


Figura 7: POUs de tipo FB y/o Programa en XML

El segundo elemento, *iec:Instances*, define el proyecto de automatización propiamente dicho.

En lo referente a la arquitectura hardware, puede describirse en dos niveles: El primer nivel está formado por nodos unidos a través de segmentos de bus, como ilustra la Figura 8.

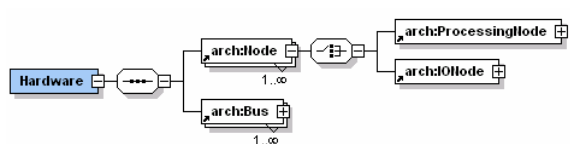


Figura 8: Estructura general de la arquitectura HW

El segundo nivel describe más concretamente la composición de los nodos que estarán formados por el mismo tipo de elementos: fuentes de alimentación, procesadores, tarjetas de entrada / salida, tarjetas de comunicación, etc. En la siguiente figura se presentan las características del nodo de procesamiento S7300 de Siemens.

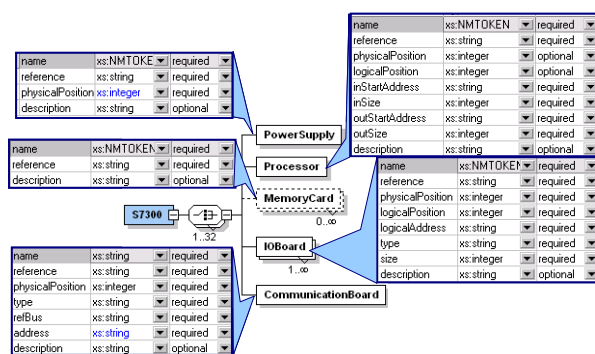


Figura 9: Nodo de Procesamiento S7300

Como ya se ha comentado la caracterización de los elementos que componen un nodo se extienden con las características propias del fabricante.

Finalmente, dado que ambas arquitecturas describen el mismo sistema, también se tienen en cuenta una serie de relaciones, ilustradas en la Tabla 3. La siguiente figura expresa en XML *schema* dichas relaciones.

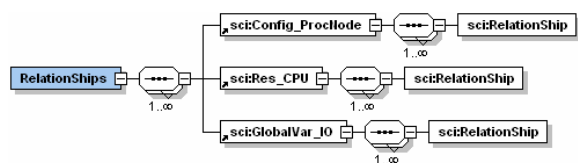


Figura 10: Relaciones entre arquitecturas en XML

3 COMPROBACIÓN DE MODELOS DE SCI

Otra tecnología XML complementaria al *schema* es el Schematron [19]. A diferencia de los *schemas*, que comprueban que una instancia XML se ajuste a una cierta gramática (énfasis en léxico, sintaxis y sistemas de tipos), el *schematron* está enfocado a la validación de contenidos (comprobación de restricciones sobre los valores y las relaciones entre los datos). Por tanto, esta tecnología es la que permite analizar la coherencia de cada una de las arquitecturas así como la del modelo completo, comprobando que la relación entre la

implementación Hardware y Software es correcta. *Schematron* se apoya en los conceptos de árbol, nodos y para búsqueda y procesamiento hace uso de XPath, un lenguaje basado en XML que permite seleccionar subconjuntos de un documento. En este sentido la tecnología *schematron* presenta una serie de ventajas frente a otros lenguajes de *schema* ya que proporciona herramientas de validación potentes empleando una sintaxis simple para su implementación.

A modo de ejemplo se presenta en la siguiente figura una regla de *schematron* encargada de comprobar que toda tarjeta de comunicaciones presente en la arquitectura hardware tiene una dirección válida.

```
<rule context="arch:CommunicationBoard">
  <assert test="@commBoardType='PROFIBUS_DP'
    and contains(@address, '.')">
    tipo de tarjeta es para conectar un segmento de bus
    Profibus y la dirección no válida
  </assert>
</rule>
```

Figura 11: Ejemplo de regla de *schematron*

De la misma manera, también son necesarias un conjunto de reglas de *schematron* para comprobación de contenidos dentro del modelo software y hardware. La siguiente figura presenta un ejemplo de regla de *schematron* para el modelo software.

```
<rule context="iec:Resource">
  <assert
    test="count(..iec:Resource)=count(..iec:Resource[@onProcessor=
      current()/@onProcessor])" priority="high">
    Los recursos de una misma configuración tienen que ir al mismo PLC
  </assert>
</rule>
```

Figura 12: Ejemplo de regla de *schematron* para SW

Finalmente, resaltar que esta tecnología es también muy útil para la comprobación de contenidos cruzados que aparecen en la verificación de las relaciones de los dos modelos (HW y SW). La Figura 13 presenta la regla de *schematron* encargada de asegurar que cada recurso del modelo software sea descargado en un procesador de un PLC de la arquitectura hardware.

```
<rule context="iec:Resource">
  <assert
    test="count(current()/[@onProcessor=//arch:IntelligentNode/@name])=1"
    priority="high">
    recurso del modelo software se tiene que descargar en un nodo
    existente en el modelo hardware
  </assert>
</rule>
```

Figura 13: Regla de *schematron* entre modelos

Por tanto, la combinación de ambas tecnologías XML se adapta perfectamente al objetivo de describir un sistema de control distribuido. En la siguiente figura se presenta el escenario general de análisis de

coherencia y consistencia de una aplicación de control industrial.

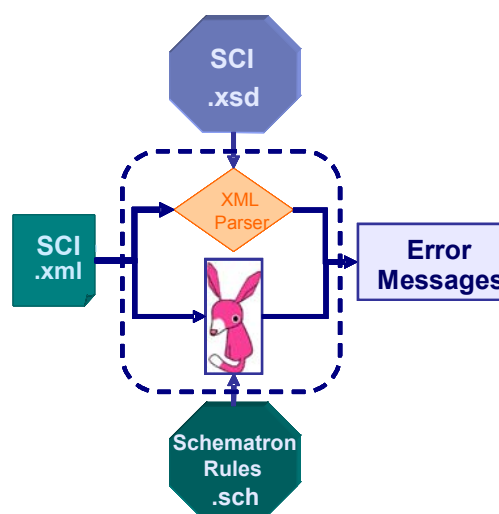


Figura 14: Escenario de análisis de coherencia y consistencia

4 INTEGRACIÓN DE HERRAMIENTAS

Para la integración de herramientas en la infraestructura, es posible utilizar diferentes tecnologías XML dependiendo de la capacidad de importar/exportar información que tenga la herramienta a integrar.

La Figura 15 ilustra los diferentes caminos posibles, así como las tecnologías XML involucradas en cada uno.

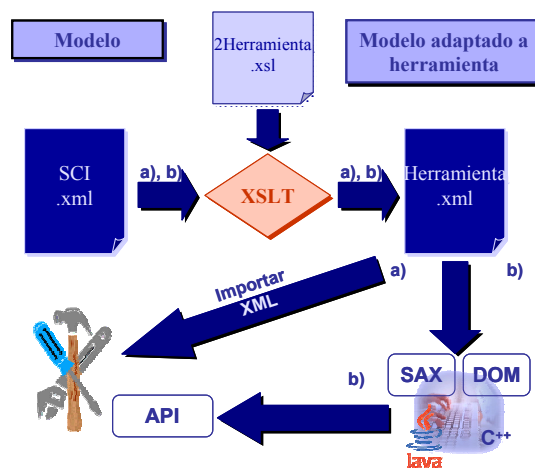


Figura 15: Tecnologías XML involucradas

En el caso de que la herramienta destinataria no tenga capacidad de exportar o importar datos, o que no lo haga en formato XML (caso b) de la Figura 15), se hace uso de las funciones de su API. Por otro lado si la herramienta exporta o importa en XML, se utilizará

directamente la opción de importar (caso *a*) de la Figura 15).

Independientemente de que el caso sea el *a*) o el *b*), se hace uso de la tecnología *eXtensible Stylesheet Language* (XSL). De esta forma, se puede añadir la información dependiente de la propia herramienta.

XSL (*eXtensible Stylesheet Language*) [5] tiene una doble funcionalidad: por un lado, es el vocabulario XML para especificar la semántica de formato de documentos de cara a la presentación. Por otro, representa el lenguaje para transformar documentos XML. En cuanto a la presentación, XSL permite generar diferentes formatos como HTML, PDF, RTF, VRML etc a partir de un único fichero XML origen. XSL permite describir la forma en que dicha información debe ser presentada. Por tanto, es una tecnología adecuada para la generación de la documentación asociada al proyecto.

Como lenguaje de transformación, XSL supone una pequeña sintaxis de lenguaje de comandos (*script*) para poder procesar los ficheros XML de forma más cómoda. Define una transformación entre un documento de entrada XML a otro de salida. Esta transformación se describe a través un conjunto de reglas. Cada regla se compone de un patrón (*pattern*) que indica en qué contexto y condiciones se disparará la regla y una acción (*template*) que indica lo que hay que realizar cuando se dispare. Esta transformación es llevada a cabo por un procesador XSL estándar atendiendo a las condiciones recogidas en la hoja de estilo (*.xsl*).

Por otro lado, las tecnologías XML **SAX** (*Simple API for XML*) [18] o **DOM** (*Document Object Model*) [4] permiten generar/manipular documentos XML en aplicaciones programadas en lenguajes de alto nivel (Java, C++, Visual Basic). Estas tecnologías junto a las XSL son las que se utilizan en el caso *b*) en la Figura 15.

En los siguientes sub-aptados se hace un estudio más en detalle de ambas situaciones. Concretamente se presentan como ejemplo herramientas de programación del proyecto de automatización.

4.1 Importar sistema desde un fichero XML

Es el caso más sencillo al que tienden hoy en día las herramientas de programación de PLCs que siguen el modelo software del estándar IEC 61131-3. En este caso, bastaría con aplicar una hoja de estilo para transformar el fichero de entrada al formato de importación de la herramienta.

A modo de ejemplo, se presenta el caso de la herramienta MULTIPROG® de KW.

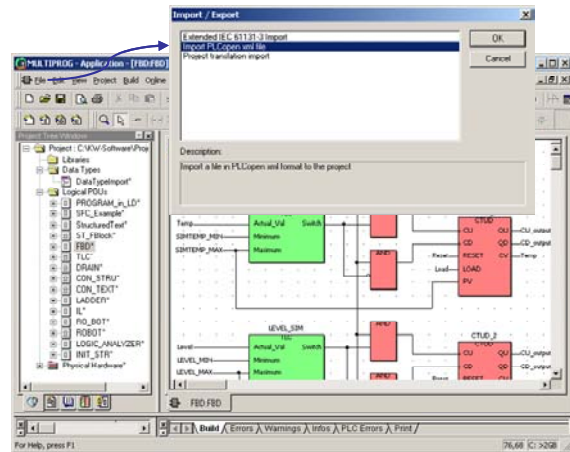


Figura 16: Ejemplo de Importar / Exportar XML

4.2 Generación del proyecto a través del API de la herramienta

En la actualidad gran parte de las herramientas de programación de PLCs no tienen la capacidad de importar / exportar proyectos de automatización de/a un documento XML, pero muchas de ellas ofrecen el servicio de exportar/importar a un documento de texto. En este caso es la tecnología XSL la encargada de procesar el modelo XML añadir las características propias de la herramienta y posteriormente dar formato de texto adecuado a ella.

La siguiente figura ilustra a modo de ejemplo la herramienta CoDeSys® de 3S Automation-Alliance.

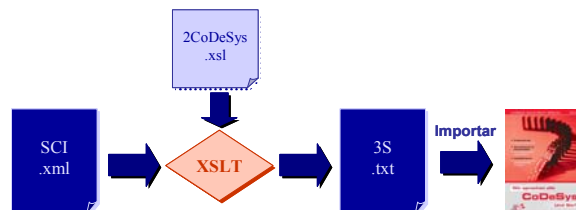


Figura 17: importar documento de texto

Si se desea exportar información de la herramienta y generar el modelo XML correspondiente, es necesario el uso de otra tecnología XML como es el DOM, que apoyada en una aplicación (desarrollada en java, c++,c#) traduzca el texto a un documento XML. Posteriormente nuevamente con una hoja de estilo (*.xsl*) se filtra la información propia de la herramienta y se le asigna la estructura/gramática acorde a la comentada en el apartado 2.

Finalmente, si la herramienta tampoco ofrece al usuario el servicio de exportar/importar el modelo a un fichero de texto con una estructura, para poder acceder y modificar información se hace a través de las funciones de su API (véase Figura 15). En la

siguiente figura se presentan las tecnologías XML involucradas con ISaGRAF Enhanced.

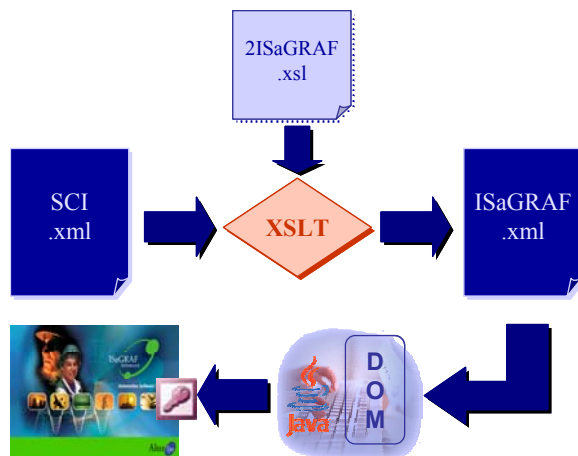


Figura 18: ISaGRAF Enhanced

En este caso la hoja de estilo *2ISaGRAF.xsl* tiene como objetivo principal añadir al *SCI.xml* la información propia de la herramienta. El fichero resultante a la transformación es *ISaGRAF.xml*.

Un proyecto ISaGRAF Enhanced® está compuesto por una base de datos MS-Access® y unos ficheros de texto con el código fuente de los POU. Estos ficheros deben almacenarse en una estructura de directorios idéntica a la estructura de configuraciones y recursos. Los ficheros se almacenarán de forma que cada carpeta correspondiente a un recurso contenga los ficheros con el código fuente de los POU contenidos en el recurso. El formato interno que utiliza la herramienta para definir un proyecto es una base de datos MS-Access®. La base de datos MS-Access se encuentra en el directorio raíz.

La aplicación desarrollada para la generación automática del proyecto de automatización parte de una base de datos MS-Access vacía que contiene la estructura de tablas utilizada por ISaGRAF Enhanced y procede a asignar a sus campos los valores definidos en el fichero *ISaGRAF.xml*

6 CONCLUSIONES

En este trabajo se ha analizado la potencialidad de las tecnologías XML en su aplicación al diseño de sistemas de automatización. Se ha demostrado, a través de la generación de prototipos, que es una tecnología abierta que puede ser utilizada para integrar las diferentes herramientas que se utilizan en el diseño de sistemas de control industrial. La aproximación propuesta por los autores se basa en la definición de modelos del sistema que son los que generan y/o consumen las herramientas y que en última instancia, son los que posibilitan la

colaboración entre los miembros del equipo de diseño. Se han presentado las diferentes tecnologías a utilizar para crear los lenguajes de dominio (XML schema), comprobar la corrección de los modelos (XML schematron), integrar herramientas (DOM, SAX, XSL), así como para generar documentación relativa al sistema en diferentes formatos (XSL-FO). Actualmente se está trabajando en la tecnología SVG para la generación gráfica de modelos del sistema

Agradecimientos

Este trabajo se ha sido subvencionado por MCYT&FEDER en el proyecto DPI 2003-2399.

Referencias

- [1] Arciniegas F. (2001) *Programación avanzada con XML*. McGraw-Hill
- [2] Bonfé, M., Fantuzzi, C. (2000) "Mechatronic Objects encapsulation in IEC 1131-3 Norm". Proceedings of the 2000 IEEE International Conference on Control Applicat., pp.598-603.
- [3] Booch, G., Rumbaugh, J., Jacobson, I.. (1999) "El lenguaje Unificado de Modelado". Addison Wesley.
- [4] DOM : www.w3.org/DOM/
- [5] Doug Tidwell (2001) *XSLT*. O'REILLY
- [6] Estévez, E., M. Marcos, U. Gangoit, D. Orive (2005). "A Tool Integration Framework for Industrial Distributed Control Systems". In Proc of the 44th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC, Seville, Spain (2005).
- [7] E. Estévez, M. Marcos, D. Orive. "Building PLC Automation Projects from IEC 61131-3 Software Architecture". In Proc of 6th IEEE Workshop on Factory Communications System Torino, Italy, (2006).
- [8] Gonzalez, V. M., F. Mateos, N. Amos (2003). MLAV. Object-Oriented Methodology for the Analysis and Modelling of the Control Logic of Discrete Event Systems, SSGRR 2003.
- [9] Heverhagen, T., Tracht, R. (2001) "Integrating UML-RealTime and IEC 61131-3 with Function Block Adapters". Proceedings of the IEEE International Symposium on Object-Oriented Real-Time Distributed Computing.
- [10] John E. Simpson (2001) *Just XML second Edition*. Prentice Hall PTR
- [11] John, K-H and Tiegkamp, M., (2001) "IEC 61131-3: Programming Industrial Automation Systems". Springer.
- [12] Kandare, G. (2001). Model-based software design for procedural process control with programmable logic controllers. The 2nd Int. PhD student workshop on systems and control, [COBISS.SI-ID 16409639].

- [13] Lewis, R.W., (1998) "Programming Industrial Control Systems using IEC 1131-3". IEE Control Engineering Series.
- [14] M. Marcos, E. Estevez. "Formal Modelling of Industrial Distributed Control Systems". 16th IFAC World Congress in Prague. 2005.
- [15] M. Marcos, E. Estevez, U. Gangoiti, I. Sarachaga, J. Barandiarán. "UML Modelling of Industrial Distributed Control Systems", CONTROL, 2004. Faro. Portugal
- [16] PLCopen (2003), Overview IEC 61131. URL: www.plcopen.org/intro_iec/overview_of_iec_61131.htm
- [17] PLCopen TC6 XML: <http://www.plcopen.com/>
- [18] SAX: <http://www.saxproject.org/>
- [19] Schematron: <http://www.ascc.net/xml/schematron/>
- [20] Thramboulidis, K. and C. Tranoris (2001). An Architecture for the Development of Function Block Oriented Engineering Support Systems. CIRA 2001.
- [21] Thramboulidis, K. (2003). An Architecture to Extend the IEC 61499 Model for Distributed Control Applications. 7th Conference on Automation Technology (Automation 2003)
- [22] Thramboulidis, K. (2004). Developing a CASE tool for distributed control Application. Int J Adv Manuf Technol (2004). 24: 24-31
- [23] Van der Vlist E. (2002) *XML Schema*. O'REILLY
- [24] XML: <http://www.w3.org/XML/>

DETECCIÓN DE LA DEGRADACIÓN DEL PROCESO WEDM BASADA EN REDES NEURONALES

E. Portillo, I. Cabanes, M. Marcos, A. Zubizarreta

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería de Bilbao, Alda. Urquijo s/n 48013 Bilbao. Tfno: 94 6013951;

FAX: 94 6014187; itziar.cabanes@ehu.es

Resumen

Este artículo presenta los resultados de un estudio comparativo de distintas configuraciones de redes neuronales aplicadas al proceso de corte por electroerosión por hilo (WEDM). En concreto, el objetivo perseguido es la detección temprana de distintas formas de comportamiento que alertan del riesgo creciente de la rotura de la herramienta empleada en este proceso de mecanizado: el hilo. Así, partiendo de un trabajo previo en el que se identificaron los diferentes tipos de fenómenos de rotura, se han entrenado distintas configuraciones de red tanto estáticas (Perceptrón Multicapa) como recurrentes (Elman). Para ello se han desarrollado tres aplicaciones en MatlabTM destinadas a la preparación de ejemplos, entrenamiento y simulación, respectivamente. La conclusión del trabajo ha sido que, si bien distintas configuraciones de red responden satisfactoriamente, la arquitectura Elman se constituye como la alternativa más viable para la detección de la degradación del proceso.

Palabras Clave: WEDM, electroerosión, ANN, redes neuronales, Perceptrón Multicapa, Elman.

1 INTRODUCCIÓN

El proceso de electroerosión por hilo (WEDM) es uno de los procesos de mecanizado no convencional más extendido en la actualidad. Este proceso es ampliamente utilizado en la fabricación de utillajes para troquelaría, que por su dureza o por sus características mecánicas resultan difíciles de mecanizar por medios convencionales. Sin embargo, uno de los problemas fundamentales del corte por electroerosión por hilo consiste en la disminución del rendimiento del proceso debida a la aparición de inestabilidades y a la rotura del hilo [1]. La dificultad en el estudio y mejora del proceso WEDM radica en que se trata de un proceso estocástico y no lineal en el que entran en juego múltiples parámetros de mecanizado.

Existen distintas líneas estratégicas en el campo de la detección de la rotura del hilo. La inmensa mayoría de los autores han desarrollado sus dispositivos de detección basándose en un único espesor de pieza [2-

6]. Todos estos autores dejan reflejado en sus publicaciones la necesidad de extender el estudio a distintos espesores, sin embargo, no existen referencias al respecto. Además, cabe destacar que, a excepción de autores como Lauwers *et al.*[3], los espesores estudiados son pequeños (20-50 mm) si se tiene en cuenta la necesidad creciente de aumentar el tamaño del espesor a mecanizar de forma óptima.

Dado que el proceso WEDM se caracteriza por ser fuertemente no lineal y estocástico, además de tener características variables con el tiempo, la descripción de modelos matemáticos que representen su comportamiento es tarea ardua. Por tanto, la utilización de técnicas de control inteligente se constituye como una opción claramente aplicable. Entre dichas técnicas destacan las técnicas heurísticas y las técnicas basadas en redes neuronales. Las primeras han sido tradicionalmente empleadas en el proceso WEDM [3], [5], [7]. Las técnicas basadas en redes neuronales han sido aplicadas en el proceso WEDM (y en otros procesos de mecanizado no convencionales: SEDM, ECM, etc) con distintos objetivos. Así, en la bibliografía se pueden distinguir tres áreas principales de aplicación: determinación de los parámetros óptimos de mecanizado [8-10]; predicción, diagnóstico y control de procesos de mecanizado no convencional [11-14] y clasificación de descargas [15-16]. Cabe destacar que en todos estos trabajos se han empleado arquitecturas de red estáticas, especialmente la Perceptrón Multicapa, para los distintos objetivos.

Como se puede observar en los antecedentes, la aplicación de técnicas basadas en redes neuronales aparece como una alternativa en boga en los procesos de mecanizado no convencionales. Sin embargo, esta no ha sido utilizada para la previsión de la rotura del hilo tal y como se plantea en el trabajo que aquí se presenta.

Tras este apartado de introducción al proceso WEDM y a la aplicación de las técnicas de redes neuronales en el mismo, en la sección 2 se presenta un resumen del trabajo que antecede al tratado en este artículo. En la sección 3 se muestra la estrategia seguida para el diseño de redes neuronales que permitan detectar regímenes de corte degradado con suficiente antelación a producirse la rotura del hilo. En la

sección 4 se describe el SW desarrollado para llevar a cabo la preparación de ejemplos, entrenamiento y simulación. En la sección 5 se analizan los resultados del estudio comparativo seleccionando el conjunto de redes más apropiado para abordar los objetivos planteados. Por último, en la sección 6 se resumen las conclusiones del trabajo.

2 IDENTIFICACIÓN DE REGÍMENES DE CORTE DEGRADADO

En el trabajo previo *Portillo et al.* [17] se presentó una metodología (resumida de forma gráfica en la Figura 1) para definir reglas heurísticas que permitieran detectar regímenes de corte degradado para así poder anticipar la rotura del hilo. En primer lugar, se obtuvo una extensa base de datos experimental mediante un sistema de adquisición que permitía capturar las señales básicas del proceso (tensión e intensidad de las descargas). Los resultados del análisis preliminar de los experimentos mostraron tres variables clave para la detección de la rotura, concretamente la energía de las descargas, la intensidad de pico y el tiempo de ionización. Así, mediante el preprocesamiento de las señales se definieron una serie de indicadores o *medidas virtuales* relacionadas con estas tres variables clave. Para ello se tomaron como referencia valores de las variables clave en régimen de corte estable. Finalmente, se definieron una serie de conjuntos de reglas heurísticas a partir del análisis del comportamiento de las medidas virtuales. Mediante dichas reglas se disparan una serie de alarmas de distintos niveles que alertan del riesgo creciente de rotura. En concreto, se identificaron tres tipos de fenómenos de rotura: incremento brusco de la energía TR-E, oscilación de la energía TR-EO, y sucesivos picos de intensidad junto con un tiempo de ionización elevado. TR-I+TDH.

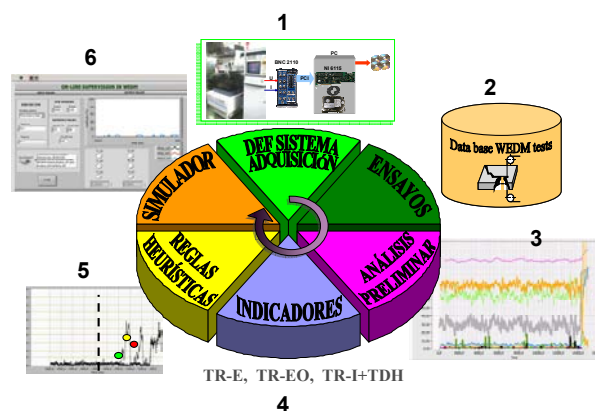


Figura 1 Metodología para la detección de la degradación del proceso WEDM

3 ESTRATEGIA BASADA EN REDES NEURONALES PARA LA DETECCIÓN DE LA DEGRADACIÓN DEL PROCESO WEDM

Tras realizar el trabajo previo en piezas de acero de espesor 50 mm y 100 mm, se plantearon dos nuevos retos:

- Establecer una metodología que permitiera predecir la rotura del hilo en espesores intermedios (entre 50 y 100 mm) a partir del comportamiento de las medidas virtuales.
- Dotar de mayor conocimiento de la calidad de las descargas al control de las máquinas de electroerosión por hilo. Concretamente, además de detectar inestabilidades y anticiparse a la rotura del hilo, se deben identificar las posibles causas que las originan. En este sentido, y dado que se ha identificado diferentes tendencias a la rotura relacionadas con un conjunto de causas, se pretende cuantificar el grado de influencia de cada una de las posibles causas en la degradación del proceso. La identificación de estas causas resulta imprescindible para abordar las futuras estrategias de actuación y mejora del proceso.

Una de las opciones que permite abordar ambos objetivos es la aplicación de técnicas basadas en redes neuronales. En este caso, las redes neuronales artificiales se presentan como una técnica muy apropiada dada la naturaleza estocástica y fuertemente no lineal del proceso WEDM. Además, esta alternativa presenta la ventaja de poder ser utilizada para atajar de manera simultánea los dos objetivos planteados.

Como se ha descrito en el apartado anterior, en un trabajo previo se ha realizado un exhaustivo análisis de la rotura del hilo caracterizándose distintos niveles de alarma y tipos de rotura. El estudio refleja que las medidas virtuales definidas aportan toda la información necesaria para reconocer una situación degradada del corte. Ante los nuevos objetivos, el planteamiento adoptado consiste en entrenar un sistema basado en redes neuronales con todo el conocimiento extraído del análisis anterior. Se trata, por tanto, de aplicar aprendizaje supervisado. Así, las entradas de la red son las medidas virtuales (que, en definitiva, son series temporales de la energía de las descargas, la intensidad de pico y el tiempo de ionización.) y el espesor de la pieza. Las salidas son los distintos niveles de alarma y el tipo de rotura.

Para poder obtener aquella red neuronal o conjunto de redes neuronales que mejor respondan a las

necesidades planteadas, se ha realizado un estudio comparativo en función de distintos criterios:

- **Arquitectura de red:** estática o recurrente. En concreto, la arquitectura de red neuronal estática analizada es la Perceptrón Multicapa PM. Esta presenta la ventaja de que es la arquitectura de red en la que existe mayor experiencia en su aplicación industrial. Por esta misma razón las herramientas SW comerciales, como MatlabTM, permiten una mayor parametrización de este tipo de redes durante su entrenamiento. En cuanto a la red recurrente, se ha utilizado la arquitectura Elman. En este tipo de redes existe claramente una menor experiencia, resultando más dificultoso su entrenamiento (definición de los targets, etc). Sin embargo, puede resultar muy beneficiosa en este caso debido a que se caracteriza por tener una naturaleza dinámica, al igual que las medidas virtuales del proceso WEDM. Esto permite reducir drásticamente el tamaño de la red ya que las series temporales de las medidas virtuales se pueden introducir punto a punto a lo largo del tiempo, sin necesidad de introducir de manera simultánea todo el histórico de datos necesario para detectar la rotura del hilo. Otra ventaja adicional sería la mayor facilidad de inclusión de este tipo de red en un sistema empotrado tanto desde el punto de vista de requisitos temporales como de requisitos de memoria.
- **Rango de los *targets* o valores deseados:** la selección del rango de los targets resulta una decisión no trivial que puede condicionar de manera importante el éxito del entrenamiento. Numerosos expertos recomiendan la aplicación de un rango simétrico $[-1, 1]$ en la arquitectura de red PM [18]. En caso de utilizar un rango asimétrico es aconsejable valorar rangos cercanos al $[0,1]$. Esto es debido al carácter asintótico de la funciones de activación sigmoideas. Por tanto, en este trabajo se han valorado los siguientes rangos: $[-1,1]$, $[0, 1]$, $[0,05, 0,95]$, $[0,1, 0,9]$, $[0,15, 0,85]$.
- **Codificación de los *targets* o valores deseados:** la codificación de los targets también se constituye como un elemento fundamental. En el caso de las redes estáticas, algunos expertos aconsejan la codificación 1-C, es decir, codificación binaria en la que se activa simultáneamente una única neurona de salida [18], [19] durante el entrenamiento. Otros contemplan la posibilidad de aplicar la codificación binaria genérica [20]. En este trabajo se valoran tanto la codificación 1-C como la binaria. En el caso de las redes recurrentes, la codificación de los targets se ha definido mediante una variedad de algoritmos llamados *IAPA (Integración Acumulativa con Pendientes Asimétricas)*. Estos algoritmos consisten

básicamente en amplificar tanto el valor de la neurona de salida como del tipo de rotura correspondiente cuando la pendiente de la medida virtual que se esté considerando aumenta. Por el contrario, si la pendiente de la forma de onda de la medida virtual en cuestión disminuye, los valores de las neuronas de salida disminuyen en menor medida. Esto facilita que las neuronas de salida adquieran valores elevados cuando se produce una oscilación en la medida virtual, identificando de este modo el tipo de rotura producido.

- **Agrupación de las entradas:** este criterio hace referencia a si se define una única red neuronal con todas las medidas virtuales como entradas (configuración simple) o si por el contrario, se definen varias redes neuronales con una o varias medidas virtuales como entradas (configuración paralela). La configuración simple únicamente se ha valorado en el caso de las redes recurrentes ya que, en el caso de las redes estáticas, el tamaño de la red sería excesivamente grande.

Como es habitual, por cada arquitectura se ha valorado distinto número de neuronas y/o capas ocultas:

- Estática: 1 ó 2 capas ocultas y entre 10 y 100 neuronas por capa oculta.
- Recurrente: entre 10 y 60 neuronas en la capa oculta.

Durante el entrenamiento se ha empleado el algoritmo de aprendizaje convencional, válido tanto para el entrenamiento de la arquitectura de red Perceptrón Multicapa PM como la arquitectura de red Elman. Se trata del algoritmo de retropropagación por descenso de gradiente con momento y tasa de aprendizaje adaptativa. Este algoritmo se caracteriza por que el error es dependiente de los valores iniciales de los pesos. Por tanto, es conveniente repetir varias veces el entrenamiento de una misma configuración de red. Así, en este trabajo se han realizado 10 iteraciones por cada configuración de red. La función de activación elegida ha dependido del rango de los targets: sigmoideal en el caso asimétrico, y tangente hiperbólica en el caso simétrico.

4 DISEÑO E IMPLEMENTACIÓN

Los programas desarrollados a lo largo del estudio tienen como objetivo facilitar tres pasos fundamentales en el entrenamiento adecuado de redes neuronales artificiales:

- la preparación de los ejemplos

- el entrenamiento de distintas configuraciones de redes neuronales, para así poder realizar el estudio comparativo
- la simulación de las redes neuronales con el fin de determinar qué estructura se adapta mejor a los objetivos planteados.

La Figura 2 muestra la metodología seguida para el diseño adecuado de las redes neuronales. Así, los programas desarrollados constituyen una herramienta versátil de entrenamiento y simulación de redes neuronales que puede ser aplicada con distintos objetivos en el proceso WEDM. Los programas mostrados en la Figura 2 han sido desarrollados en MatlabTM. Estos son descritos en los apartados que se presentan a continuación.

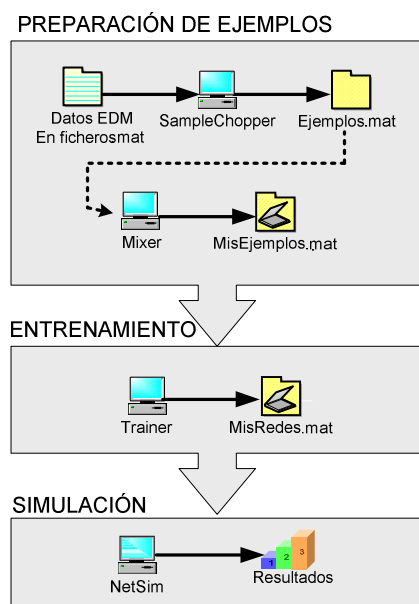


Figura 2 Esquema general del diseño de ANN

4.1. SW DE GENERACIÓN DE EJEMPLOS

Una de las etapas más importantes en cualquier proceso de aplicación de redes neuronales artificiales es la selección de los ejemplos de entrenamiento y validación. En este caso, los ejemplos se han obtenido a partir de la exhaustiva base de datos experimental obtenida en el trabajo previo. Así, las señales tensión e intensidad de las descargas, contenidas en la base de datos, son procesadas obteniendo como resultado las medidas virtuales. Estas son: energía, intensidad de pico, tiempo de ionización elevado y tiempo de ionización corto [17]. Concretamente, los ejemplos de entrenamiento y validación contienen:

- Alarmas de tres niveles: nivel bajo, nivel medio y/o nivel alto de riesgo de rotura.

- Tipos de rotura: incremento brusco de la energía TR-E, oscilación de la energía TR-EO, y sucesivos picos de intensidad junto con un tiempo de ionización elevado TR-I+TDH.

En total se disponen de 153 ejemplos. El 70% se ha utilizado para el entrenamiento y el 30% para validación. Para llevar a cabo el tratamiento de los ejemplos se ha desarrollado el programa *SampleChopper* en MatlabTM. Esta aplicación permite visualizar los posibles ejemplos de entrenamiento y validación, seleccionar aquellos que se consideran adecuados y guardarlos en archivos binarios tras generar sus targets (para los distintos rangos planteados en la estrategia). La Figura 3 muestra la interfaz de usuario de *SampleChopper*. La ventana principal permite la selección y almacenamiento de los ejemplos y viene acompañada de una ventana flotante que facilita la previsualización de los targets generados por el programa.

La interfaz de usuario se estructura en 4 paneles, cuya funcionalidad se detalla a continuación:

- Gráfica principal: muestra la secuencia de puntos de las cuatro medidas virtuales.
- Panel de errores: es un listado de errores y advertencias sobre acciones erróneas del usuario.
- Listado de ejemplos: permite imprimir un listado de los ejemplos cargados o añadidos.
- Control A1, A2 y A3: estos controles permiten indicar (manual o automáticamente) el instante de tiempo en el que se produce cada una de las alarmas.
- Control datos entrada: permite navegar por los ejemplos cargados.
- Variables temporales: permite guardar los datos actuales en un archivo intermedio con el fin de no perder los datos de una sesión en caso de fallo del ordenador.
- Control fichero salida: indica en qué fichero se desea guardar los ejemplos seleccionados. El usuario puede guardar los ficheros en diferentes formatos: bien con un rango de targets concreto o bien con una arquitectura de red concreta.

Una vez generados los ficheros de ejemplos con la aplicación *SampleChopper*, el *Mixer* permite obtener un paquete único de ejemplos en el que se encuentran claramente identificados los ejemplos de entrenamiento por un lado, y los de validación por el otro.

4.2. SW DE ENTRENAMIENTO

Como se ha comentado anteriormente, el núcleo de la estrategia propuesta consiste en realizar un estudio comparativo en función de distintos criterios:

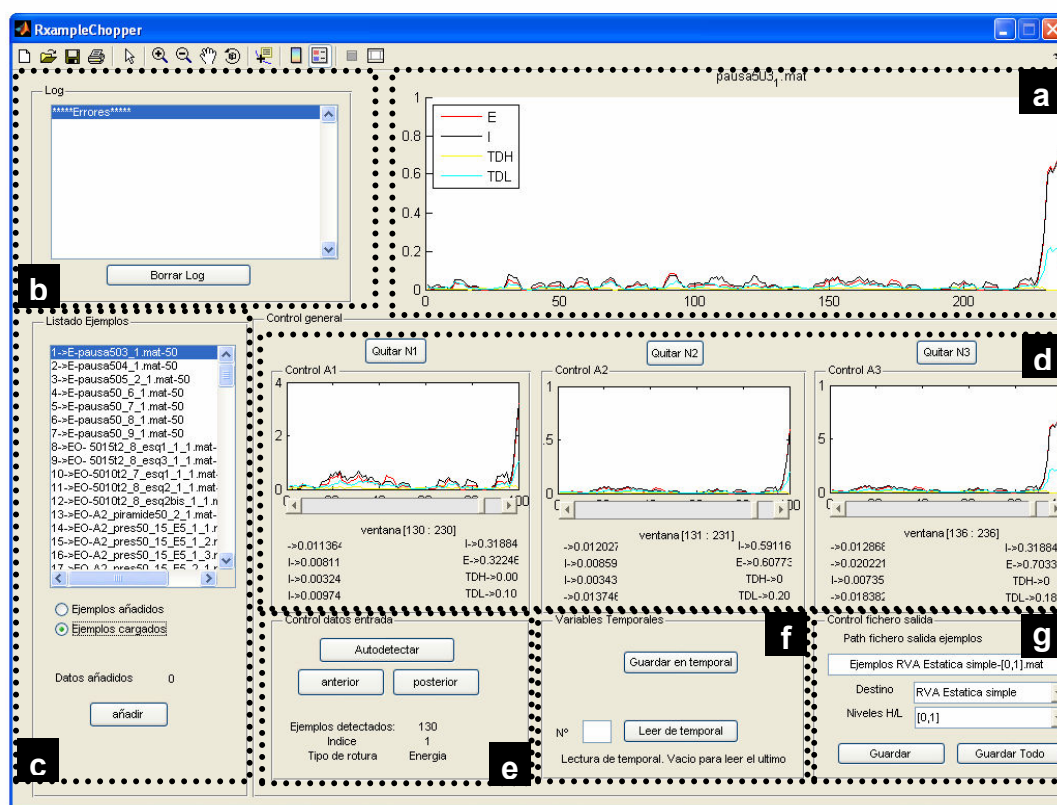


Figura 3. Interfaz de usuario de SampleChopper

- Arquitectura de red: estática (PM) o recurrente (Elman).
- Rango de los *targets* o valores deseados: $[-1,1]$, $[0, 1]$, $[0,05, 0,95]$, $[0,1, 0,9]$, $[0,15, 0,85]$.
- Codificación de los *targets* o valores deseados: codificación 1-C o binaria en estáticas, y codificación IAPA-1, IAPA-2 o IAPA-3 en redes recurrentes.
- Agrupación de las entradas: configuración simple o configuración paralela.

Para poder analizar las distintas alternativas, se ha desarrollado una aplicación de entrenamiento en MatlabTM llamada *Trainer* (ver Figura 4).

Durante el entrenamiento se ha empleado el algoritmo de retropropagación convencional. El entrenamiento de una determinada red finaliza cuando se cumple una de las siguientes condiciones:

- Se alcanza el error objetivo.
- Se alcanza un número determinado de épocas de entrenamiento.
- El error de validación aumenta durante un número determinado de épocas consecutivas.

La Figura 4 muestra la interfaz de usuario del *Trainer*. Esta consta de 3 paneles en los que se resumen las funcionalidades del programa, y que se explican a continuación:

- a) Cargar Estructura: este panel permite cargar el listado de estructuras que contienen los ejemplos generados en el *Mixer*.
- b) Datos: muestra información acerca del paquete cargado: el tipo de red al que está destinado, el número de ejemplos que contiene y el rango de los *targets*.
- c) Control Entrenamiento: permite al usuario el control de los parámetros de entrenamiento. También permite escalar los *targets* al rango $[-1,1]$. El control *Número iteraciones* se refiere al número de veces que se a va a realizar el entrenamiento de cada configuración de red.

Una vez finalizado el entrenamiento, se guarda de manera automática el fichero que contiene la estructura que almacena el objeto red resultado del entrenamiento, así como una serie de campos que permiten identificar y caracterizar la red (ejemplos de entrenamiento y validación, errores de entrenamiento y validación, etc).

4.3. SW DE SIMULACIÓN

Aunque el *Trainer* ofrece datos acerca del error de entrenamiento y validación, estos datos sólo sirven para realizar una comparación entre los diferentes tipos de red. Por esta razón, el SW de simulación *NetSim* aporta una medida real de cómo va a funcionar la red, es decir, en qué grado es capaz de detectar situaciones de corte degradado. El objetivo del programa *NetSim*, por tanto, es la simulación de la red utilizando los ejemplos de entrenamiento y validación (u otros nuevos) mostrando, ejemplo a ejemplo, las salidas reales que produce la red ya entrenada.

La interfaz de usuario de *NetSim* se divide en 4 paneles (ver Figura 5):

- Cargar Red: este panel permite cargar los paquetes de las redes ya entrenadas.
- Datos: da información diversa acerca de los parámetros de entrenamiento así como de la configuración de las redes del paquete cargado.
- Simulación: permite ejecutar la simulación sobre la red seleccionada en el panel anterior.
- Navegación: Lista todos los ejemplos de entrenamiento y validación que se han simulado. Si se hace clic en ellos se muestra una ventana mostrando las salidas que corresponden a cada ejemplo y los targets o valores deseados.

Los datos obtenidos mediante *NetSim* son de gran ayuda para poder conocer el funcionamiento real de una determinada configuración de red. La utilidad más importante aportada por la aplicación es la visualización por cada ejemplo de los patrones, salidas reales y targets. Esto último es lo que verdaderamente permite medir en qué medida es útil la red evaluada para los objetivos planteados.

5 RESULTADOS

En esta sección se muestran los resultados del estudio realizado. Así, durante la fase de análisis de resultados el objetivo ha sido identificar la red neuronal o conjunto de redes neuronales que resulten más adecuadas en la detección tanto de los niveles de alarma como del tipo de rotura.

5.1. REDES ESTÁTICAS: PERCEPTRÓN MULTICAPA

La Tabla 1 muestra el mejor resultado obtenido en el caso de una red paralela que procesa la medida virtual de energía. En este caso la entrada de la red está constituida por 100 puntos de la medida virtual de la energía (que se corresponden con un histórico

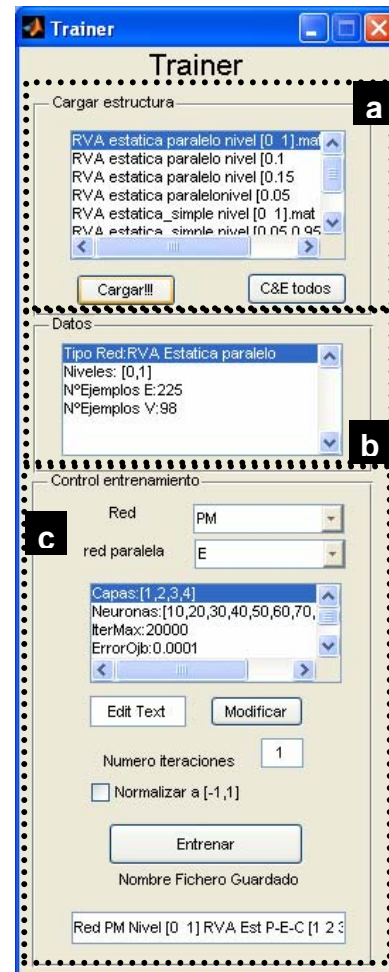


Figura 4 Interfaz de usuario del *Trainer*

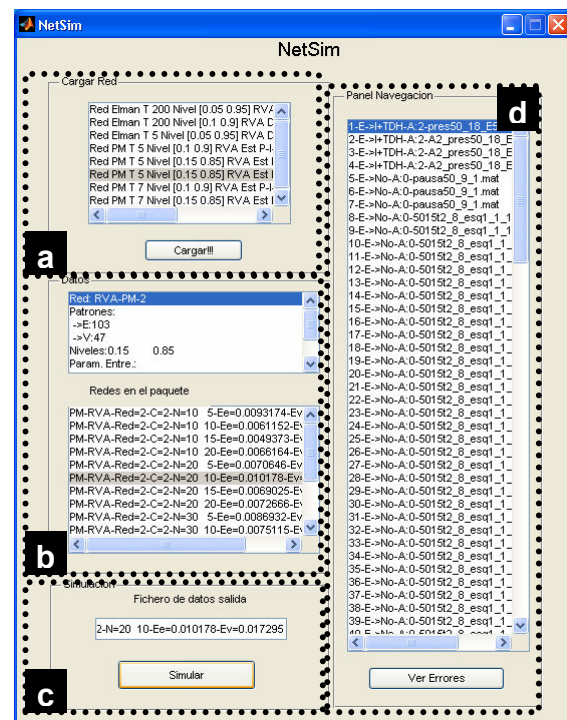


Figura 5 Interfaz de usuario de *NetSim*

de 100 ms) y una neurona que indica el espesor de la pieza. La salida está formada por cinco neuronas: tres de ellas corresponden a los niveles de alarma A1, A2 y A3, y dos de ellas indican los tipos de rotura TR-E y TR-EO.

Arquitectura	Codificación	Nivel targets	Error validación
PM 101-10-15-5	Binaria	[0.15,0.85]	MSE=0.0332

Tabla 1 Mejores resultados en PM Energía

La Tabla 2 muestra la configuración de red paralela de intensidad que presenta los resultados más satisfactorios. De manera similar, la entrada de la red está constituida por 100 neuronas en las que se introducen 100 ms de la medida virtual de la intensidad y una neurona que indica el espesor de la pieza. La salida está formada por tres neuronas: tres de ellas indican dos niveles N1 y N2, y una de ellas indica el tipo de rotura TR-I+TDH.

En este caso se habla de niveles en general (y no de *niveles de alarma*) porque los niveles son relevantes siempre y cuando el tiempo de ionización elevado TDH sea significativo.

Arquitectura	Codificación	Nivel targets	Error validación
PM 101-10-3	Binaria	[0.1, 0.9]	MSE= 0.0106

Tabla 2 Mejores resultados en PM Intensidad

Varias configuraciones han dado lugar a resultados satisfactorios. Entre dichas configuraciones, la mejor es la presentada en la Tabla 3.

Al igual que en los dos casos anteriores, la entrada de la red está constituida por 100 neuronas en las que se introducen 100 ms de la medida virtual del tiempo de ionización elevado TDH y una neurona que indica el espesor de la pieza. La salida depende del tipo de codificación aplicado. En el caso de codificación binaria, la salida está formada por una única neurona, que indica si el nivel de TDH es alto o no.

Arquitectura	Codificación	Nivel targets	Error validación
PM 101-40-20-1	Binaria	[0,1]	MSE=1.15 70e-004

Tabla 3 Mejores resultados en PM TDH

Como se puede observar en la Tabla 3, los resultados han sido plenamente satisfactorios. Por tanto, se debería considerar la selección de una arquitectura más sencilla (de una sola capa) ya que es aceptable un mayor error de validación.

A modo de ilustración, la Figura 6 muestra un ejemplo de validación con TR-EO y TR-E respectivamente. En las gráficas aparecen los

patrones o ejemplos de validación. En la parte superior del marco de la figura se encuentran indicados los valores deseados o targets como, por ejemplo, $T: 0,15 \ 0,15 \dots$. Sobre la gráfica aparecen los valores de salida reales, $S: 0,05029 \ 0,11015 \dots$. Como se puede apreciar, las salidas han sido activadas conforme a los valores deseados.

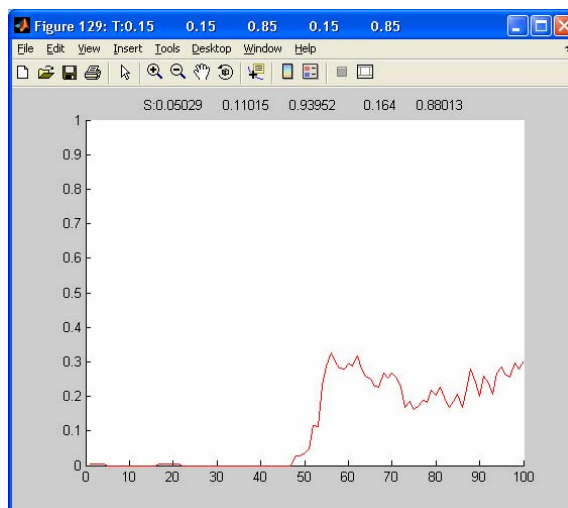


Figura 6 Ejemplo de validación con mejor caso en la alternativa PM Energía: TR-EO

5.2. REDES RECURRENTE: ELMAN

La Tabla 4 muestra la mejor configuración obtenida en la red paralela Elman de energía. Debido a que se trata de un tipo de arquitectura de red recurrente, el número de neuronas de entrada es tres (dos de ellas se corresponden con la medida virtual de la energía y de la intensidad respectivamente, y la otra con el espesor de la pieza). Esta red dispone de tres salidas: una de ellas indica el nivel de alarma N, y las dos restantes permiten identificar los tipos de rotura TR-E y TR-EO respectivamente. En este caso, al añadir como entrada la medida virtual de la intensidad, se ha conseguido una mejor diferenciación de los tipos de rotura.

Arquitectura	Codificación	Nivel targets	Error validación
Elman 10	IAPA-1	[0,1,0,9]	MSE=8.5 10 ⁻⁴

Tabla 4 Mejores resultados en Elman energía

La Tabla 5 muestra la mejor configuración obtenida en la red paralela Elman de intensidad. La red dispone de dos entradas que se corresponden con la medida virtual de la intensidad y con el espesor de la pieza respectivamente. Esta red también tiene tres neuronas de salida: una de ellas indica el nivel de alarma, y la neurona restante permite identificar el tipo de rotura TR-I+TDH.

Arquitectura	Codificación	Nivel targets	Error validación
Elman 10	IAPA-1	[0.05,0.95]	MSE=0.0012

Tabla 5 Mejores resultados en Elman intensidad

En la Tabla 6 aparece la configuración de red seleccionada de entre todas las redes entrenadas de tipo paralelo Elman de tiempo de ionización elevado TDH. Si bien no es la configuración con menor MSE de todas las obtenidas, es la que mejor relación tamaño/rendimiento presenta. De nuevo, la red presenta dos neuronas de entrada: una para la medida virtual TDH, y otra para el espesor de la pieza. La salida está formada por una única neurona, que permite reconocer si el nivel de TDH es significativo o no.

Arquitectura	Codificación	Nivel targets	Error validación
Elman 10	IAPA-3	[0.1,0.9]	MSE=0.0088

Tabla 6 Mejores resultados en Elman TDH

A modo de ejemplo, la Figura 7 muestra un ejemplo de validación con TR-EO. En la gráfica superior de la figura aparecen las entradas de la red (Patrones). La gráfica situada en el medio muestra la salida real de la red. La gráfica situada en la parte inferior muestra los valores deseados o targets, que en este caso se han codificado con el algoritmo IAPA-1. Como se puede apreciar, las neuronas de salida N y TR-EO siguen adecuadamente al target, mientras la neurona TR-E permanece en nivel bajo.

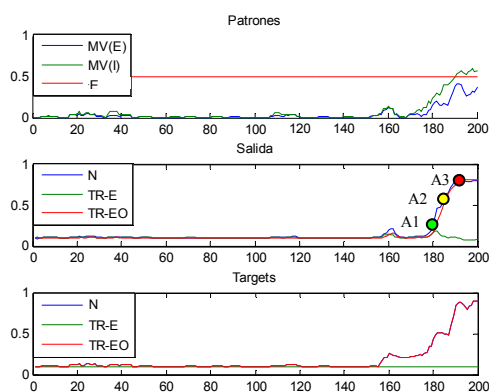


Figura 7 Ejemplo de validación con mejor caso en la alternativa Elman Energía: TR-EO

5.3. CONFIGURACIÓN DE LAS REDES NEURONALES SELECCIONADAS

Tras el estudio comparativo realizado, las configuraciones de red que mejor responden a los objetivos planteados han sido las basadas en la

arquitectura Elman. Además, la aplicación de esta arquitectura se justifica también desde el punto de vista de requisitos temporales y de requisitos de memoria. Así, la arquitectura recurrente permite en este caso obtener tamaños de red significativamente menores. Además, evita tanto el coste computacional como de desarrollo que implicaría la gestión del buffer de las entradas en una arquitectura estática.

La Figura 8 muestra el esquema general del sistema de detección de inestabilidades en WEDM basado en ANN. Como se puede observar, está constituido por tres redes neuronales Elman en paralelo: red paralela de la energía, red paralela de intensidad y red paralela de tiempo de ionización elevado TDH.

El post-procesamiento de las salidas de la red consiste básicamente en lo siguiente:

- Post-procesamiento del nivel de alarma: activación de las alarmas que superan un determinado umbral establecido durante la etapa de análisis de resultados.
- Post-procesamiento del tipo de rotura: cuantificación del grado de influencia de cada una de los tipos de rotura en la degradación el proceso.

6 CONCLUSIONES

En este trabajo se ha demostrado que la aplicación de redes neuronales en la detección de la degradación del proceso WEDM se constituye como una alternativa claramente viable. Esta alternativa, además, presenta la ventaja de que tiene visos de responder a dos objetivos simultáneamente: predecir la rotura del hilo en espesores intermedios y cuantificar el grado de influencia de cada una de las posibles causas de rotura en la degradación el proceso. Otro aspecto fundamental a tener en cuenta es que el proceso WEDM se caracteriza por una dinámica muy rápida. Un sistema basado en redes neuronales, por tanto, se presenta como una técnica altamente apropiada para ser integrada en un sistema de control en tiempo real dada la rapidez computacional de las redes neuronales.

Tras la elaboración de un estudio comparativo de distintas configuraciones de red en arquitecturas estáticas (PM) y recurrentes (Elman), se ha observado que esta última es la más apropiada dada la naturaleza dinámica de los patrones y targets. Además, esta arquitectura permite obtener tamaños de red significativamente menores si se comparan con PM, facilitando así su aplicación en tiempo real.

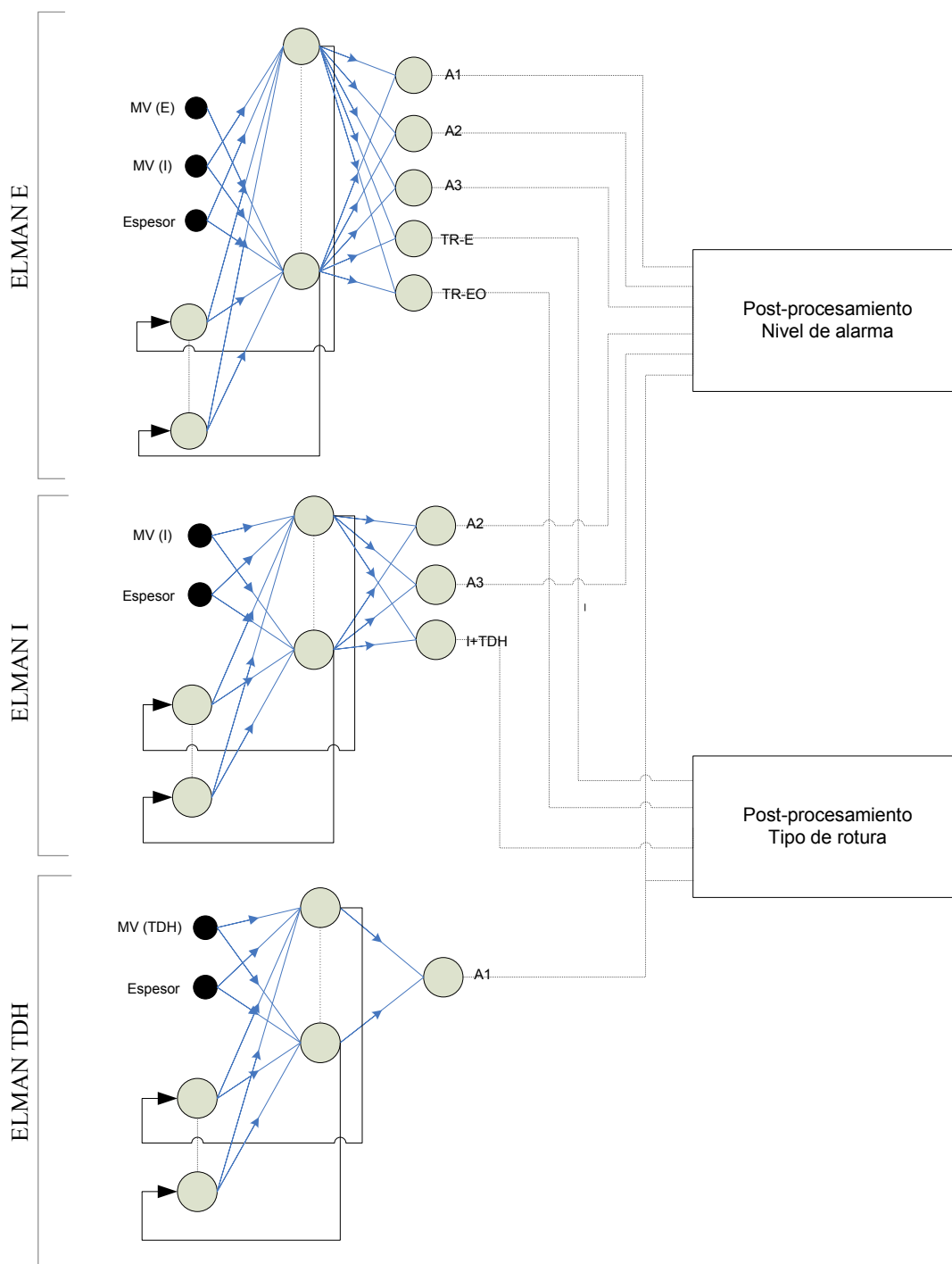


Figura 8 Esquema general del sistema de detección de inestabilidades en WEDM basado en ANN

Agradecimientos

Este trabajo ha sido subvencionado por la Universidad del País Vasco a través del proyecto UPV05/114 Aplicación de Técnicas de Control Inteligente en el Proceso de Electroerosión por Hilo, así como por el Departamento de Educación, Universidades e Investigación del Gobierno Vasco (beca predoctoral BFI04.383).

Referencias

- [1] Ho K. H., Newman S. T., Rahimifard S., Allen R. D. *State of the art in wire electrical discharge machining (WEDM)*. International Journal of Machine Tools and Manufacture, Vol. 44, pp. 1247-1259, 2004.
- [2] Wu, J and Li, M.H., *The identification of the servo control state in wire electrical discharge*

- machining process*, ISEM XIII, pp. 423-433, 2001.
- [3] Lauwers, B.; Kruth, J.P.; Bleys, Ph.; Van Coppenolle, B.; Stevens, L. y Derighetti, R., *Wire Rupture Prevention Using On-Line Pulse Localisation in WEDMB*. VDI Berichte NR. 1405, 1998.
- [4] Obara, H.; Abe, M. y Ohsumi, T. *Control of Wire Breakage during Wire EDM*. International Journal of Electrical Machining, No. 4, 1999.
- [5] Yan, M.T. and Liao, Y.S., *Adaptive control of WEDM process using the fuzzy control strategy*, ISEM XI, pp.343-352, 1995.
- [6] Shoda, K.; Kaneko, Y.; Nishimura, H.; Kunieda, M.; Fan, M.X., *Development of Adaptive Control System to prevent EDM wire breakage*. EDM Technology, Vol. 3, pp. 17-22, 1995.
- [7] Shoda, Kaneko, Nishimura, Kunieda, Fan, *Development of Adaptive Control System to prevent EDM wire breakage*. EDM Technology, Vol. 3, pp. 17-22, 1995.
- [8] Tarng, Y.S., Ma, S.C. y Chung, L.K., *Determination of optimal cutting parameters in WEDM*, International Journal of Machine Tools and Manufacture, Vol. 35, No. 12, pp. 1693-1701, 1995.
- [9] Gi-Nam Wang, *Two-Phase Reverse Neural Network Approach for Modeling a Complicate Manufacturing Process with Small Sample Size*, Neural Information Processing, Vol. 2, No. 1, January 2004.
- [10] Cao Fenggou, Yang Dayong, *The study of high efficiency and intelligent optimization system in EDM sinking process*, Journal of Materials Processing Technology, 149, 83-87, 2004.
- [11] Kuo-Ming Tsai, Pei-Jen Wang, *Predictions on surface finish in electrical discharge machining based upon neural network models*, International Journal of Machine Tools & Manufacture, 41, pp. 1385-1403, 2001.
- [12] J.T. Huang y Y.S. Liao, *A wire edm maintenance and fault-diagnosis expert system integrated with an artificial neural network*, Int. J. Prod. Res., Vol. 38, No. 5, pp. 1071-1082, 2000.
- [13] Yan, Liao, y Chang, *On-line Estimation of Workpiece Height by using Neural Networks and Hierarchical adaptive Control of WEDM*. J. Advanced Manufacturing Technology, p.p. 884-89, 2001.
- [14] Behrens, A. and Ginzel, J., *Neuro-Fuzzy Process Control System for Sinking EDM*, Journal of Manufacturing processes, vol. 5, nº 1 pp. 33-39, 2003.
- [15] J. Y. Kao and Y. S. Tarng, *A neural-network approach for the on-line monitoring of the electrical discharge machining process*, Journal of Materials Processing Technology, Vol. 69, 1-3, p.p. 112-119, 1997.
- [16] T.K.K.R. Mediliyegedara, A.K.M. De Silva,, D.K. Harrison, J.A. McGeough, *An intelligent pulse classification system for electro-chemical discharge, machining (ECDM)—a preliminary study*, Journal of Materials Processing Technology, 149, pp. 499-503, 2004.
- [17] Cabanes, I., Portillo, E., Marcos, M; Orive, D. and Sánchez, J.A., *Simulador para la detección de inestabilidades en el proceso WEDM basado en reglas heurísticas*, XXVI Jornadas de Automática. ISBN: 84-689-0730-8, 2005.
- [18] <ftp://ftp.sas.com/pub/neural/FAQ.html>
- [19] <http://www.dontveter.com/bpr/data.html>
- [20] Pedro Isasi Viñuela, Inés M. Galván León, *Redes de Neuronas Artificiales. Un enfoque práctico*, Ed. Pearson Educación, S.A., Madrid, 2004, ISBN: 84-205-4025-0.

Aplicación de la Síntesis de Lyapunov Borrosa a un caso práctico real: la grúa porta-contenedores

Andújar Márquez, J. M., Barragán Piña, A. J.

Ctra. Huelva-Palos de la Frontera S/N, andujar@diesia.uhu.es, antonio.barragan@diesia.uhu.es

Gegúndez Arias, M. E., Maestre Hachero, M.

Ctra. Huelva-Palos de la Frontera S/N, gegundez@dmu.uhu.es, maestre@dmu.uhu.es

Resumen

En este trabajo se presenta una metodología que, extendiendo la síntesis clásica de Lyapunov al dominio de las palabras, permite obtener de forma sistemática la base de reglas de un controlador borroso así como una expresión analítica para la ley de control resultante.

Palabras clave: Estabilidad, controlador, lógica borrosa, computación con palabras.

1 INTRODUCCIÓN

Las críticas usuales al uso de la lógica borrosa en el diseño de sistemas de control ha sido su escasa formalización matemática –las soluciones son a menudo empíricas– y la ausencia de herramientas estándares formalizadas que eviten que cada problema sea abordado con una metodología diferente.

Los autores de este trabajo están interesados en la formalización del modelado y el control borroso [1, 2, 3], de modo que las herramientas y metodologías de análisis y síntesis aceptadas por la comunidad científica sean de aplicación al modelado y control borroso [2, 4, 5]

El paso más importante (y el más complicado) en el diseño de controladores borrosos es la construcción de la *base de reglas* [6]. Este trabajo está basado en un nuevo método para diseñar la base de reglas de los controladores borrosos, mediante la extensión del método de síntesis de Lyapunov clásico al dominio del *cálculo con palabras* [7]. Como punto de partida se supone que está disponible alguna información lingüística que describe el comportamiento de la planta. Esto es naturalmente una suposición realista pues tal información puede ser obtenida bien desde una comprensión física de la planta o por la intervención de un operador experto sobre la misma.

La lógica borrosa permite que en base al conocimiento anterior, sea construido un modelo lingüístico del controlador sin necesidad de disponer de un modelo matemático de la planta,

el cual muchas veces es o muy difícil de desarrollar o su complejidad es tal que lo hace impráctico.

Sin embargo, la pregunta que surge es ¿cómo podemos usar esta información lingüística sobre la planta (y, naturalmente, el control objetivo especificado) para construir sistemáticamente la base de reglas para un controlador borroso apropiado?

Es bien sabido que el modelado y el control en el marco clásico (no borroso) está basado en dos pasos:

- Construir un modelo *matemático* de la planta
- Usar algún enfoque de síntesis como el de Lyapunov, para diseñar un controlador que la estabilice.

En el caso que nos ocupa, el primer paso es reemplazado por:

- Construir un modelo *lingüístico* de la planta

Para el segundo paso, se ha de transformar el enfoque de síntesis clásico desde el dominio de las ecuaciones matemáticas al dominio de las lingüísticas, esto es, hacer *cálculos con palabras* [12]. Esto se realizará extendiendo la síntesis clásica de Lyapunov al dominio de las palabras. Este nuevo enfoque es denominado *Síntesis de Lyapunov Borrosa*.

Básicamente, se sigue el método clásico de síntesis de Lyapunov mediante la construcción de la función de Lyapunov *candidata* V . A continuación se determinan las condiciones que se requieren para hacer en *efecto* a V una función de Lyapunov del sistema en lazo cerrado. Esto hace que, supuesto *conocimiento borroso* sobre la planta que hay que controlar, las condiciones a cumplir puedan ser formuladas como *reglas borrosas* *Si-Entonces*. Este conjunto de reglas borrosas constituyen la base de reglas para el controlador borroso.

Este método puede ser usado para derivar la base de reglas sistemáticamente. Además, el hecho que la base de reglas se diseñe usando funciones de Lyapunov, tiene implicaciones importantes para

el posterior análisis de estabilidad del controlador (que no será tratado en este trabajo), ya que al final, la metodología que se presenta, permite obtener una expresión analítica para la ley de control.

2 Método de Diseño

Se comenzará describiendo la síntesis clásica de Lyapunov, usada cuando se tiene una descripción matemática de la planta. Posteriormente esta síntesis clásica será extendida al caso borroso.

Considérese el sistema autónomo

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (1)$$

donde $\mathbf{f}(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot))^T$, las $f_i(\cdot)$ son funciones continuas, $\mathbf{x} = (x_1 x_2 \dots x_n)^T$ es el vector de estado del sistema y $\mathbf{u} = (u_1 u_2 \dots u_p)^T$ es la entrada de control (T designa la trasposición).

El objetivo de control es estabilizar el sistema alrededor de un punto de trabajo \mathbf{x}_0 (sin pérdida de generalidad se puede considerar que $\mathbf{x}_0 = \mathbf{0}$). Más específicamente: Diseñar una ley de control $\mathbf{u}(\mathbf{x})$, tal que $\mathbf{0}$ sea un punto de equilibrio estable de (1). Definiciones alternativas de estabilidad (por ejemplo, estabilidad asintótica, estabilidad exponencial) pueden ser usadas.

Cuando se conoce el modelo exacto (1), un camino para alcanzar este objetivo es elegir una función de Lyapunov *candidata* $V(\mathbf{x}(t))$ y determinar las condiciones sobre \mathbf{u} (nótese que V depende de \mathbf{x} y en consecuencia de \mathbf{u}) necesarias para que V sea en efecto una función de Lyapunov (esto es, condiciones que garantizan que $\dot{V} \leq 0$).

Sin embargo, desde el punto de vista de la lógica borrosa, el problema de control puede ser abordado suponiendo que el modelo matemático exacto (1) es *desconocido* pero se tiene algún conocimiento parcial borroso sobre él, por ejemplo, tal como

SI x_1 es *grande* y u_2 es *pequeño* ENTONCES f_1 es *grande*.

El objetivo de este artículo es utilizar para el sistema borroso, un método similar al clásico para diseñar el controlador, esto es, elegir una función *candidata* V , obtener una expresión para su derivada \dot{V} , y entonces derivar condiciones sobre \mathbf{u} tal que V pueda ser una función de Lyapunov de hecho. Es importante notar sin embargo que, como tenemos sólo conocimiento borroso del sistema, el requisito $\dot{V} \leq 0$ se convierte en una ecuación *lingüística* cuya solución origina *condiciones borrosas* para \mathbf{u} , que pueden estar en una de dos posibles representaciones. Según el modelo borroso la primera forma es del tipo Mandani [11]:

SI x_1 es $\langle v1 \rangle$ y/o x_2 es $\langle v1 \rangle \dots$ y/o x_n es $\langle v1 \rangle$
ENTONCES u_i debe ser $\langle v1 \rangle$.

donde $\langle v1 \rangle$ son variable lingüísticas (por ejemplo, *grande*, *pequeño*). La segunda forma es del tipo Takagi-Sugeno (TS) [9]:

SI x_1 es $\langle v1 \rangle$ y/o x_2 es $\langle v1 \rangle \dots$ y/o x_n es $\langle v1 \rangle$
ENTONCES u_i debe ser $g(x_2, x_2, \dots, x_n)$.

donde $g(\cdot)$ es una función de sus variables.

A esta forma de computar con palabras las condiciones del teorema de estabilidad de Lyapunov para sintetizar controladores que estabilicen sistemas borrosos se le denomina *Síntesis de Lyapunov Borrosa*.

3 Un caso práctico: grúa porta-contenedores

Las grúas porta-contenedores son usadas en muchos puertos para cargar y descargar contenedores hacia y desde barcos. Los contenedores quedan fijados mediante cables flexibles montados en la cabecera de la grúa. La cabecera de la grúa de mueve sobre una guía horizontal. Cuando se levanta un contenedor y la cabecera de la grúa comienza a moverse, el contenedor empieza a balancearse. Este balanceo, si bien no es un problema durante el transporte del contenedor, sí lo es para poder soltarlo ya que para poder alojar el contenedor en el barco, éste ha de care *a plomo*.

Existen dos soluciones triviales para resolver este problema [10]. Una es posicionar la cabeza de la grúa exactamente sobre la posición de descarga, y entonces, esperar hasta que el contenedor deje de balancearse hasta un nivel aceptable. Si el día no es ventoso, es posible llegar a esta situación, aunque ciertamente consume mucho tiempo. Obviamente, por razones de coste, un barco porta-contenedores ha de ser cargado y descargado en el menor tiempo posible. La otra solución consiste en elevar el contenedor y desplazarlo muy lentamente para que no se balancee. Nuevamente, si el día no es ventoso, es posible realizar esta operación, aunque ciertamente a coste de emplear mucho tiempo en la misma. Una solución alternativa consiste en disponer de contenedores con cables adicionales para su fijado durante la operación de carga y descarga. Sin embargo, esta solución, no es muy habitual ya que es bastante costosa.

Existen operadores humanos expertos que a partir del control continuo de velocidad del motor de la cabeza de la grúa, compensan el balanceo con avances y retrocesos hasta posicionar el con-

tenedor en el sitio exacto. Esta operación, si bien no es fácil, sí es realizada de forma aceptable por gruístas expertos.

La solución pasa por la automatización de este problema de control no es trivial, ya que los modelos de planta obtenidos [8] son sensiblemente no lineales y pueden alcanzar grados de complejidad muy elevados: el comportamiento del motor de la cabeza de la grúa es no lineal, la cabeza de la grúa se mueve con fricción, las perturbaciones tales como rachas de viento son imprevisibles, etc.

Sin embargo, a pesar de todos los problemas mencionados, está claro que un operador experto es capaz de controlar la grúa sin necesidad de utilizar complejas ecuaciones diferenciales. Por tanto, parece éste un problema apropiado para ser abordado desde el punto de vista de la lógica borrosa.

Para automatizar el control de la grúa, se emplean sensores para la posición de la cabeza de la grúa (variable **Distancia**) y el ángulo de balanceo del contenedor (variable **Ángulo**). Éstas serán las variables de entrada que configurarán los antecedentes de las reglas. La variable de salida sería la potencia del motor, la cual configurará el consecuente de las reglas. El modelo borroso a elaborar es del tipo Mandani, esto es, la potencia será un conjunto borroso.

Los posibles valores de las variables lingüística [10] (denominados *términos* o *etiquetas*) son: para la variable **Distancia** {lejos, media, cerca, cero, negativo_cerca} donde que la distancia sea *negativa* nos indica que nos hemos pasado de la posición objetivo; para la variable **Ángulo** {pos_grande, pos_pequeño, cero, neg_pequeño, neg_grande} y para la **Potencia** {pos_alta, pos_media, cero, neg_media, neg_alta} donde la potencia *negativa* nos indica la potencia a aplicar en el caso que la distancia sea negativa (ver Figuras 1, 2 y 3).

Sea x la variable **Distancia** y α la variable **Ángulo**.

Sea $V = \frac{1}{2}(x^2 + \alpha^2)$ la función de Lyapunov candidata.

Para que el controlador sea estable, se ha de cumplir que $\dot{V} = x\dot{x} + \alpha\dot{\alpha} \leq 0$.

Para ello se ha de tener en cuenta que:

- (1) La **Potencia** del motor es proporcional a la distancia (en sentido opuesto).
- (2) La **Potencia** del motor es proporcional a la velocidad con que se desplaza el contenedor sobre el brazo de la grúa.
- (3) La velocidad del **Ángulo** es proporcional a la

Potencia que suministremos al motor.

En base a las premisas anteriores, podría definirse una base de reglas, basadas en el sentido común aplicado al conocimiento que se tiene de la planta, para implementar el control de la grúa. Esta base de reglas, elaborada completamente con palabras, podría ser la siguiente:

- R1: Si x es *Lejos* y α es *cero* entonces debe ocurrir $\dot{x} < 0$ (fuerza inicial, media), **Potencia** es *pos_media*.
- R2: Si x es *Lejos* y α es *negativo_pequeño* $\dot{x} < 0$ debe ser *negativo_grande* y, (ya que el ángulo es negativo) y, en consecuencia, **Potencia** es *positiva_alta*.
- R3: Si x es *media* y α es *negativo_pequeño* entonces \dot{x} ha de ser *negativo_media*. Como el ángulo es negativo-pequeño, hay que elegir **Potencia** es *positiva_alta*.
- R4: Si x es *media* y α es *negativo_grande* entonces \dot{x} es *negativo_medio* y en consecuencia, **Potencia** es *positiva_media*.
- R5: Si x es *cerca* y α es *positivo_pequeño* entonces **Potencia** es *negativa_media*.
- R6: Si x es *cerca* y α es *cero* entonces **Potencia** es *negativa_media*.
- R7: Si x es *cerca* y α es *negativo_pequeño* entonces **Potencia** es *positiva_media*.
- R8: Si x es *cero* y α es *positivo_pequeño* entonces **Potencia** es *negativa_media*.
- R9: Si x es *cero* y α es *cero* entonces **Potencia** es *cero*.

Para cada variable lingüística, cada término se define mediante sus funciones de pertenencia.

Para la variable **Distancia**, x , se definen las funciones de pertenencia siguientes (ver Figura 1):

$$\mu_{\text{negativo_cerca}}(x) = [1 + e^{2k_1(x+c_1/3)}]^{-1}$$

$$\mu_{\text{cero}}(x) = e^{-kx^2}$$

$$\mu_{\text{cerca}}(x) = e^{-k(x-c_1)^2/2}$$

$$\mu_{\text{media}}(x) = e^{-k(x-2c_1)^2/2}$$

$$\mu_{\text{lejos}}(x) = [1 + e^{-2kc_1(x-8c_1/3)}]^{-1}$$

Para la variable **Ángulo**, α , se definen las siguientes funciones de pertenencia (ver Figura 2):

$$\mu_{\text{neg_grande}}(\alpha) = [1 + e^{k_2c_2(\alpha+3c_2/2)}]^{-1}$$

$$\mu_{\text{neg_pequeño}}(\alpha) = e^{-k_2(\alpha+c_2)^2/4}$$

$$\mu_{\text{cero}}(\alpha) = e^{-k_2\alpha^2}$$

$$\mu_{\text{pos_pequeño}}(\alpha) = e^{-k_2(\alpha-c_2)^2/4}$$

$$\mu_{\text{pos_grande}}(\alpha) = [1 + e^{-k_2c_2(\alpha-3c_2/2)}]^{-1}$$

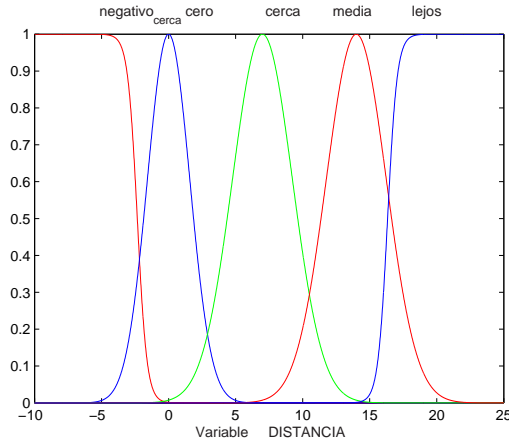


Figura 1: Funciones de pertenencia para la variable de entrada *Distancia*

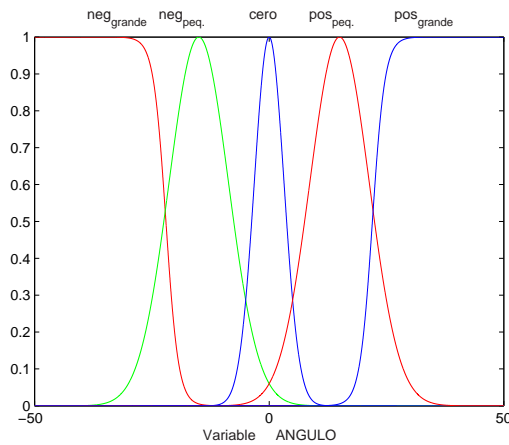


Figura 2: Funciones de pertenencia para la variable de entrada *Ángulo*

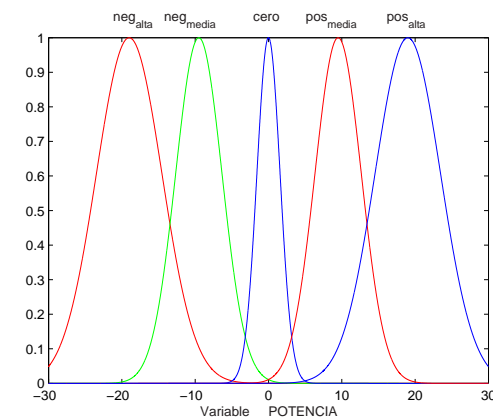


Figura 3: Funciones de pertenencia para la señal de control (variable *Potencia*)

Por último, la señal de control, definida por la variable u , queda implementada por las siguientes funciones de pertenencia:

$$\mu_{\text{neg_alta}}(u) = e^{-k_3(u+2c_3)^2/4}$$

$$\mu_{\text{neg_media}}(u) = e^{-k_3(u+c_3)^2/2};$$

$$\mu_{\text{cero}}(u) = e^{-2k_3u^2}$$

$$\mu_{\text{pos_media}}(u) = e^{-k_3(u-c_3)^2/2}$$

$$\mu_{\text{pos_alta}}(u) = e^{-k_3(u-2c_3)^2/4}$$

donde $c_1, c_2, c_3, k_1, k_2, k_3 \in \mathbb{R}^+$.

Puesto que al final del proceso de inferencia se ha de actuar sobre el motor, la señal de control ha de adquirir en cada momento un valor concreto (*crisp*), esto es, tantos kW. Por tanto, el sistema de control necesita desborrosificar la variable lingüística *Potencia* para producir un valor real.

En el sistema borroso desarrollado se aplica el método de inferencia producto y como desborrosificador el centro-promedio. Esto es

$$\vec{u}(x, \alpha) = \frac{\sum_{i=1}^4 w_i u_i}{\sum_{i=1}^4 w_i} \quad (2)$$

donde w_i es el grado de cumplimiento de la regla i -ésima.

Esta fórmula reúne tanto el método de inferencia como de desborrosificación.

u_i es el valor para el que se produce el valor máximo del conjunto borroso de salida en la regla correspondiente. Por ejemplo, en la regla R1, u es “positiva media”, $\mu_{\text{positivo_media}}(u) = 1 \Leftrightarrow u_1 = -c_3$, y así sucesivamente para todas las reglas).

Introduciendo todos estos valores en la expresión 2 se tiene:

$$\vec{u}(x, \alpha) = \frac{\text{num}(x, \alpha)}{\text{den}(x_1, x_2)}$$

donde

$$\begin{aligned} \text{num}(x, \alpha) = & \mu_{\text{lejos}}(x) \cdot \mu_{\text{cero}}(\alpha) \cdot c_3 \\ & + \mu_{\text{lejos}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \cdot 2c_3 \\ & + \mu_{\text{media}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \cdot (2c_3) \\ & + \mu_{\text{media}}(x) \cdot \mu_{\text{neg_grande}}(\alpha) \cdot (c_3) \\ & + \mu_{\text{cerca}}(x) \cdot \mu_{\text{pos_pequeño}}(\alpha) \cdot (-c_3) \\ & + \mu_{\text{cerca}}(x) \cdot \mu_{\text{cero}}(\alpha) \cdot (-c_3) \\ & + \mu_{\text{cerca}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \cdot (c_3) \\ & + \mu_{\text{cero}}(x) \cdot \mu_{\text{pos_grande}}(\alpha) \cdot (-c_3) \\ & + \mu_{\text{cero}}(x) \cdot \mu_{\text{cero}}(\alpha) \cdot 0 \end{aligned}$$

$$\begin{aligned}
\text{den}(x, \alpha) = & \mu_{\text{lejos}}(x) \cdot \mu_{\text{cero}}(\alpha) \\
& + \mu_{\text{lejos}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \\
& + \mu_{\text{media}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \\
& + \mu_{\text{media}}(x) \cdot \mu_{\text{neg_grande}}(\alpha) \\
& + \mu_{\text{cerca}}(x) \cdot \mu_{\text{pos_pequeño}}(\alpha) \\
& + \mu_{\text{cerca}}(x) \cdot \mu_{\text{cero}}(\alpha) \\
& + \mu_{\text{cerca}}(x) \cdot \mu_{\text{neg_pequeño}}(\alpha) \\
& + \mu_{\text{cero}}(x) \cdot \mu_{\text{pos_pequeño}}(\alpha) \\
& + \mu_{\text{cero}}(x) \cdot \mu_{\text{cero}}(\alpha)
\end{aligned}$$

Para valores particulares de las distintas constantes que aparecen descritos a continuación, se han obtenido dos expresiones que aproximan en la medida que se indica a la expresión del controlador. Así para los valores $k_1 = 0'2$, $k_2 = 0'05$, $k_3 = 0'1$, $c_1 = 7$, $c_2 = 15$ y $c_3 = 9'5$, se han obtenido para el controlador las siguientes aproximaciones:

$$\begin{aligned}
u(x, \alpha) = & 6'9037 \tanh(0'3725x - 4'9822) \\
& - 5'6503 \tanh(0'1609\alpha + 0'7204) \\
& + 10'2853 \tanh(0'0220x - 0'0328\alpha + 0'7820)
\end{aligned} \quad (3)$$

que proporciona un **error Medio** = 3.2468.

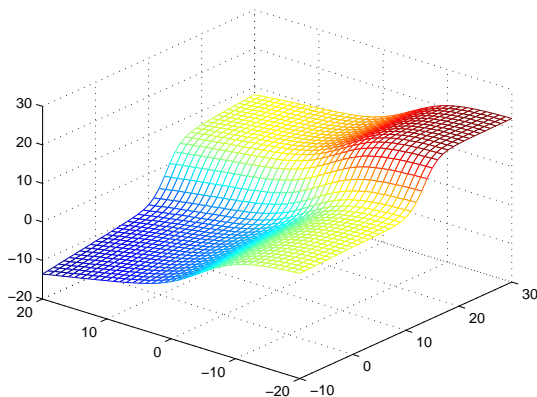


Figura 4: Gráfica de la función $u(x, \alpha) = 6'9037 \tanh(0'3725x - 4'9822) - 5'6503 \tanh(0'1609\alpha + 0'7204) + 10'2853 \tanh(0'0220x - 0'0328\alpha + 0'7820)$

Como segunda aproximación:

$$\begin{aligned}
u(x, \alpha) = & 673'7822 \tanh(0'0009x + 0'0004) \\
& - 8'4893 \tanh(0'1080\alpha + 0'3474)
\end{aligned} \quad (4)$$

que proporciona un **error Medio** = 4.1065.

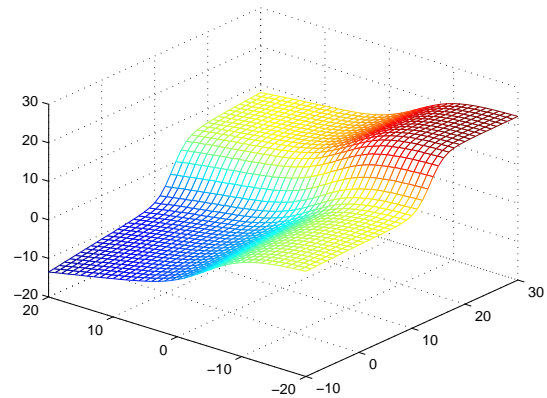


Figura 5: Gráfica de la función $u(x, \alpha) = 673'7822 \tanh(0'0009x + 0'0004) - 8'4893 \tanh(0'1080\alpha + 0'3474)$

4 Conclusiones

En este trabajo se ha presentado una primera aproximación a una metodología que, extendiendo la síntesis clásica de Lyapunov al dominio de palabras, permite obtener de forma sistemática la base de reglas de un controlador borroso así como una expresión analítica para la ley de control resultante.

Este resultado permite dar un paso en la formalización de los sistemas de control borroso, ya que al disponerse de una expresión analítica se pueden abordar metodologías de análisis clásicas.

Agradecimientos

Este trabajo es una contribución del proyecto DPI2005-01065 financiado por el Ministerio de educación.

Referencias

- [1] Andújar, J. M., Aroba, J., De La Torre, M. L., Grande, J., (2006) "Contrast of evolution models for agricultural contaminants in ground waters by means of fuzzy logic and data mining", *Environmental Geology*, **49**, nº 1, pp. 458-466.
- [2] Andújar, J.M., Bravo, J. M., Peregrín, A., (2004), "Stability Analysis and synthesis of multivariable fuzzy systems using interval arithmetic", *Fuzzy Sets and Systems*, **148**, nº 3, pp. 337-353.
- [3] Andújar, J.M., Bravo, J. M., (2005), "Multivariable fuzzy control applied to the physical-chemical treatment facility of a cellulose factory", *Fuzzy Sets and Systems*, **150**, nº 3, pp. 475-492.

- [4] Andújar, J.M., Barragán, A. J., (2005), “A Methodology to design stable nonlinear fuzzy control systems”, *Fuzzy Sets and Systems*, **154**, n° 2, pp. 157-181.
- [5] Andújar, J.M., Barragán, A. J., Córdoba, J.M., Fernández, I., (2006), “Diseño de sistemas de control borroso: Modelado de la planta”, *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*, **3**, n° 1, pp. 75-81.
- [6] Lee, C. C., (1990) “Fuzzy logic in control systems: Fuzzy logic controller, (Parts I and II),” *IEEE Trans. Systems, Man, Cybernetics*, **20**, n° 2, pp. 404–435.
- [7] Margaliot, M., Langholz, G., (2000) “New approaches to fuzzy modeling and control: design and analysis”, *Series in Machine Perception Artificial Intelligence*, World Scientific, Singapore.
- [8] Sousa, J. M., Babuška, R., Bruijin, B., Verbruggen, H., (1996), “Comparison of conventional and fuzzy predictive control”, in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, pp. 1782–1787.
- [9] Takagi, T., Sugeno M. (1985). “Fuzzy identification of systems and its applications to modeling and control”, *IEEE Trans. Systems, Man, Cybernetics*, **15**, n° 1, pp. 116-132.
- [10] Von Altrock, C., (1995), “Fuzzy Logic & neurofuzzy applications explained”, Prentice-Hall.
- [11] Yager, R. R., Filev, D. P: (1994). “Essentials of Fuzzy Modeling and Control”, John Wiley & Sons.
- [12] Zadeh, L. A. (1996), “Fuzzy logic = computing with words”, *IEEE Trans. Fuzzy Systems*, **4**, n° 2, pp. 103-111.

SIMULACIÓN DEL DIAGRAMA DE RADIACIÓN ULTRASÓNICO MODIFICADO POR BOCINAS Y VALIDACIÓN EXPERIMENTAL DEL MODELO DE ELEMENTOS FINITOS

Mónica Fernández Moreno

Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática
E.T.S. de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria
Avda. Los Castros s/n, 39005, Santander, Cantabria, ESPAÑA
E-mail: monica@teisa.unican.es

C. Rodríguez, L. Alonso, J. Pérez Oria

Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática
E.T.S. de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria
Avda. Los Castros s/n, 39005, Santander, Cantabria, ESPAÑA
E-mail: [cristina,alonso,oria]@teisa.unican.es

Resumen

Teniendo en cuenta la influencia que los elementos mecánicos acoplados a sensores ultrasónicos tipo bocinas poseen sobre el lóbulo de radiación, surge la necesidad de encontrar un método de simulación que permita encontrar las características óptimas de dicho elemento con el fin de mejorar el patrón de radiación para una determinada aplicación. En este trabajo se presenta la validación experimental sobre el eje de radiación de la simulación de un modelo mediante la técnica de elementos finitos.

Palabras Clave: ultrasonidos, bocina, lóbulo de radiación ultrasónico, elementos finitos

un mayor alcance para una determinada frecuencia de operación. Las técnicas que permiten dicha tarea son la utilización de arrays de sensores, que implica una gran complejidad hardware, o la utilización de elementos conformadores como son las bocinas, que no aumentan la complejidad del sistema.

A la vista de los resultados obtenidos en trabajos anteriores [1] y [2] con el uso de elementos conformadores, se vio la necesidad de obtener un modelo que permitiera obtener por simulación las características deseadas del lóbulo de radiación mediante el uso de elementos mecánicos. Es ahí donde aparece la idea de crear y validar experimentalmente un modelo simulado.

1 INTRODUCCIÓN

La utilización de sensores ultrasónicos en aire es de diversa y variada aplicación en la industria, y su objetivo barre desde tareas elementales como son medida de distancias o detección de objetos, hasta tareas de complejidad media o alta como son la detección de defectos en procesos de producción [6], o incluso sistemas de reconocimiento o identificación de objetos [5]. La principal limitación de los sensores ultrasónicos radica en que al aumentar la frecuencia de operación se acorta la distancia de trabajo. Se añade además el hecho de que la anchura del lóbulo de radiación depende asimismo de dicha frecuencia. Para mejorar las características direccionales en una determinada aplicación es necesario aumentar la frecuencia de operación, lo que provoca una disminución en la distancia de trabajo. Se plantea entonces la tarea de encontrar nuevas técnicas de conformación de lóbulo de radiación que impliquen

2 DESCRIPCIÓN DEL SISTEMA

En este trabajo se distinguen dos partes bien diferenciadas. La primera de ellas es la obtención del lóbulo de radiación ultrasónico mediante simulación empleando la técnica de elementos finitos. Esta simulación comprende tanto la radiación del sensor en el espacio libre como con la utilización de elementos conformadores.

La segunda parte se basa en la obtención del lóbulo de radiación ultrasónico de forma experimental.

Para la primera parte del trabajo se optó por un modelo de sensor ultrasónico que se corresponde con el de un pistón mecánico con forma de disco. Este pistón está encargado de transmitir un movimiento vibratorio (tanto longitudinal como transversal) a partículas de materiales contiguos y suponiendo que la superficie del mismo vibra con una amplitud y fase constantes.

Cuando se utiliza el método de elementos finitos, se ha de resolver la ecuación de Helmholtz que caracteriza la propagación de una onda estacionaria:

$$\Delta p(\vec{r}) + k^2 p(\vec{r}) = 0 \quad \text{en } \Omega \quad (1)$$

Donde Δ es el operador laplaciano, p es la presión acústica, \vec{r} es la posición, k es el número de onda y Ω es el dominio de trabajo.

Para realizar la simulación se han usado coordenadas cilíndricas, de forma que el sistema se reduce a un problema a 2 dimensiones. Dada la simetría del problema se ha considerado el *eje y* como eje de simetría de manera se reduce el espacio de simulación.

En la Figura 1 se muestra el dominio de trabajo usado para el caso de radiación libre.

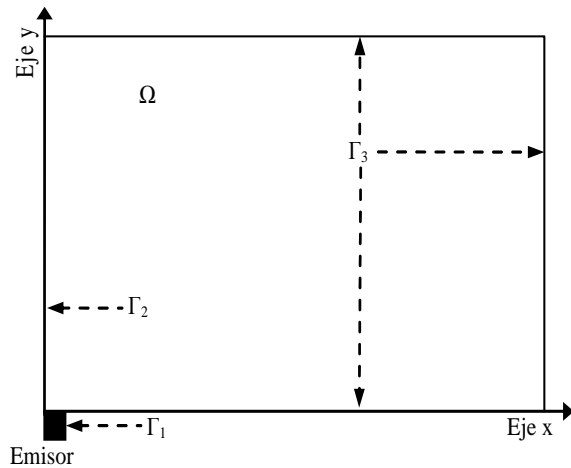


Figura 1: Dominio de trabajo para la radiación libre

En este dominio de trabajo se tienen las siguientes condiciones de contorno:

$$p(\vec{r}) = 1 \quad \text{en } \Gamma_1 \quad (2)$$

$$\frac{\partial p}{\partial n} = 0 \quad \text{en } \Gamma_2 \quad (3)$$

$$\frac{\partial p}{\partial n} = ikp \quad \text{en } \Gamma_3 \quad (4)$$

Donde \vec{n} es la normal unitaria hacia afuera, i es la unidad imaginaria, Ω es el dominio de trabajo, Γ_1 es la superficie del emisor, Γ_2 es el eje de simetría y Γ_3 es la frontera del dominio.

La ecuación 2 es la condición de contorno de Dirichlet, que determina la presión sobre la superficie del emisor, para nuestro caso constante y de valor

unitario; la ecuación 3 representa la condición de contorno del eje de simetría; la ecuación 4 es la condición de contorno de Newmann, que indica que la propagación de las ondas en dichas paredes se realiza sin reflexión.

En la Figura 2 se muestra el caso de la simulación realizada con bocina, en la que las condiciones de contorno sobre Γ_1 , Γ_2 y Γ_3 son las mismas que en el caso de radiación libre, mientras que la condición de contorno en la bocina viene dada por la ecuación 5, en la que la derivada de la presión respecto de la normal en la superficie de la bocina es nula.

$$\frac{\partial p}{\partial n} = 0 \quad \text{en } \Gamma_4 \quad (5)$$

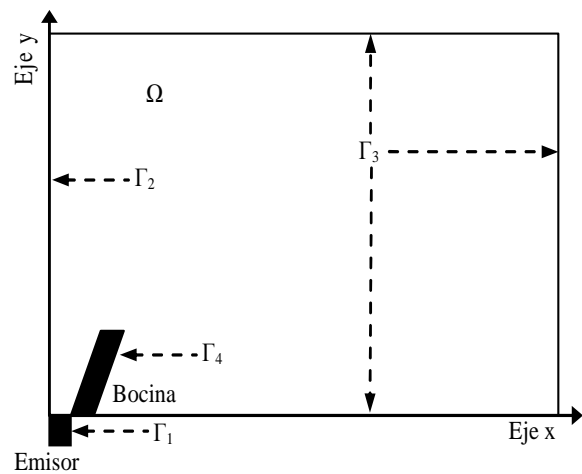


Figura 2. Dominio de trabajo con bocina

Debido a la necesidad de validar experimentalmente el modelo simulado, en la segunda parte del trabajo se obtuvo el lóbulo de radiación ultrasónico de manera experimental. Debido a que para gran parte de las aplicaciones es interesante estudiar la presión acústica sobre el eje de radiación, se optó especialmente por realizar las medidas sobre dicho eje.

Para dichas medidas se han usado dos transductores (de la marca Hexamite), uno funcionando como emisor y otro funcionando como receptor, ambos con una frecuencia de trabajo de 40KHz. Además del correspondiente acondicionador de señal (proporcionado por el mismo fabricante). El conjunto emisor-receptor utilizado puede verse en la Figura 3.

Durante la realización de las medidas la distancia entre emisor y receptor fue variada desde 16cm. a 1m. con un paso de 1 cm. Además, para la adquisición y tratamiento de las señales, se ha empleado una tarjeta de adquisición de datos de 10 Msamples/sg.



Figura3: Sistema emisor-receptor con elemento conformador

Para la realización de las medidas con elemento conformador se han usado bocinas de geometría variada, presentándose en este trabajo una bocina cuya características geométricas se detallan en la Figura 4 y en la cual el *eje y* es el eje de simetría.

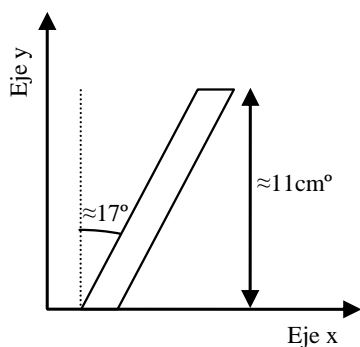


Figura4: Geometría del elemento conformador

3 ANÁLISIS DE LAS SIMULACIONES

Las simulaciones realizadas se llevaron a cabo teniendo en cuenta varios factores que afectan al lóbulo de radiación, como son el radio del diámetro del sensor y el coeficiente de atenuación del aire que depende de la temperatura y humedad entre otros factores.

Atendiendo únicamente al tamaño del sensor, se obtienen lóbulos de radiación distintos, como se puede observar en la Figura 5 y Figura 6, para el caso de radiación libre.

En la Figura 5, se presenta la simulación para un sensor radio 0.5 cm. y trabajando a una frecuencia de 40 KHz., mientras que en la Figura 6, lo que se representa es el mismo caso pero con un sensor de radio 1cm. Se comprueba que si se aumenta el

tamaño del sensor, se aumenta la directividad del lóbulo de radiación, como era de esperar.

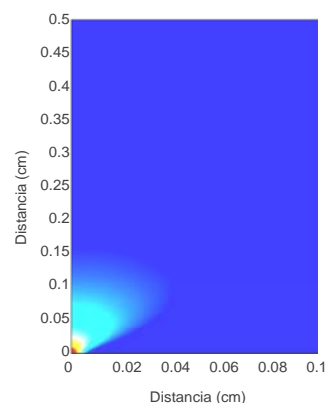


Figura 5. Simulación con radiación libre y radio sensor 0.5cm.

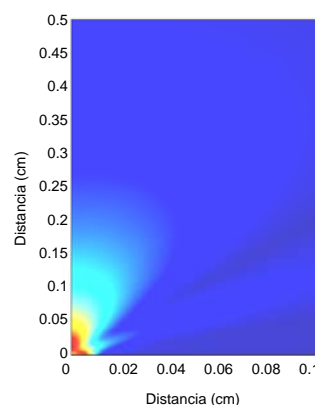


Figura 6. Simulación con radiación libre y radio del sensor de 1cm.

Si se analiza el efecto del tamaño del sensor en el caso de simulación usando elemento de conformador se obtienen las Figuras 7 y Figura 8, en las que los resultados son los mismos que en el caso de radiación sin bocina.

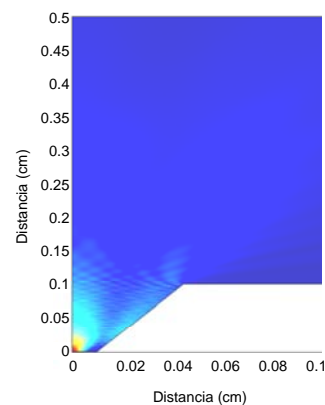


Figura 7 Simulación con elemento mecánico y radio sensor 0.5cm.

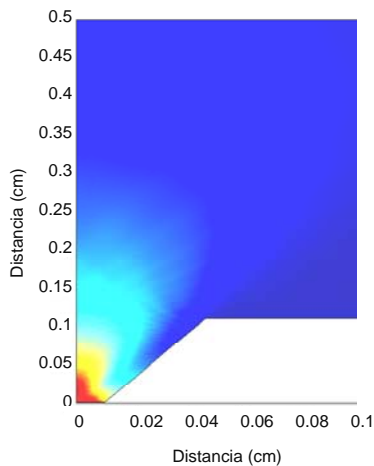


Figura 8 Simulación con bocina y radio sensor 0.5cm.

Aunque si el estudio se centra sobre el eje de radiación, se observa que el tamaño del sensor no afecta a la distancia de trabajo, ya que la relación de presiones acústicas entre curvas es prácticamente constante. Ver Figura 9 y 10.

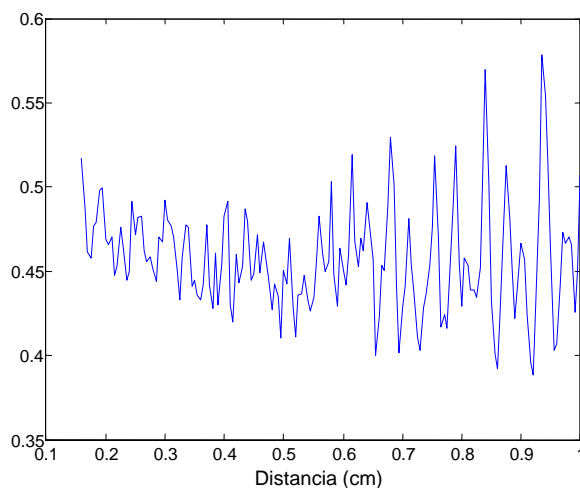


Figura 9: Relación entre presiones acústicas sobre el eje de radiación para 2 tamaños de sensor distintos y en el caso de radiación libre.

En la Figura 9, esta relación está entorno al valor de 0.5 que es la relación que existe entre el tamaño de los sensores. Por otro lado cabe destacar, que las oscilaciones que se producen para las distancias grandes se deben a que la simulación con elementos finitos requiere que el número de ecuaciones que hay que resolver aumenta de una manera exponencial al aumentar el campo de trabajo, y por lo tanto, se requiere una gran potencia hardware para obtener mayor resolución. Para nuestro caso, fue preferible tener un mayor alcance (hasta 1m. sobre el *eje y*) aunque se disminuyera algo la resolución de la simulación.

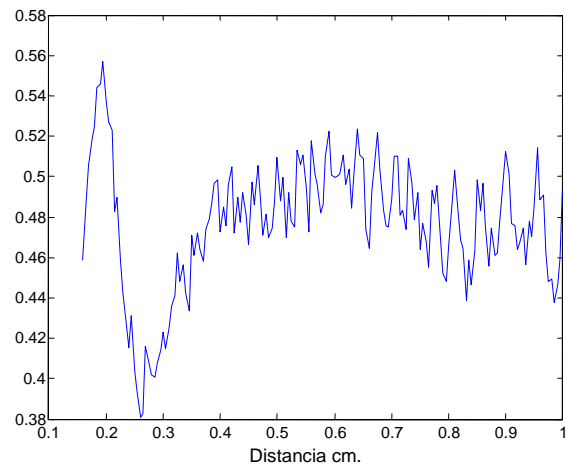


Figura 10: Relación entre presiones acústicas sobre el eje de radiación para 2 tamaños de sensor distintos y en el caso con bocina.

Para la Figura 10, se observa la misma relación (aproximadamente sobre el valor de 0.5), exceptuando a pequeñas distancias ya que en este caso la presión acústica se ve influenciada debido a que a cortas distancias con el uso del elemento conformador se produce un fenómeno de interferencias de las ondas en las zonas cercanas al elemento conformador.

Si se atiende al coeficiente de atenuación se observa que al aumentar el coeficiente de atenuación, en el caso de la no utilización de elemento conformador, se acorta la distancia de trabajo, como es lógico pensar a priori. Ver Figura 11.

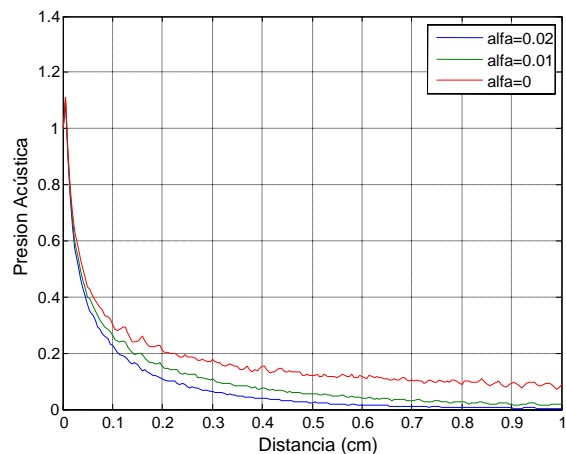


Figura 11. Comparación de la simulación en radiación libre para tres coeficientes de atenuación distintos.

Si analizamos el mismo efecto del coeficiente de atenuación para el caso de simulación con elemento conformador se obtiene la Figura 12, en la que los resultados coinciden con los de la simulación en el espacio libre.

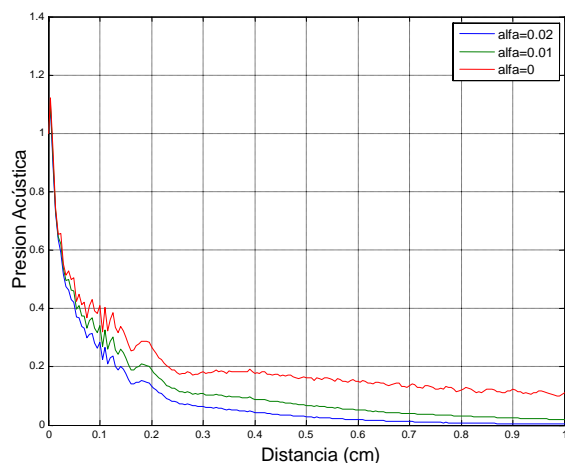


Figura 12. Comparación de la simulación con bocina para tres coeficientes de atenuación distintos.

A la vista de las simulaciones con radiación libre y con elemento conformador, se observa que con la utilización de éste último se aumenta la distancia de trabajo, como puede verse en la Figura 13.

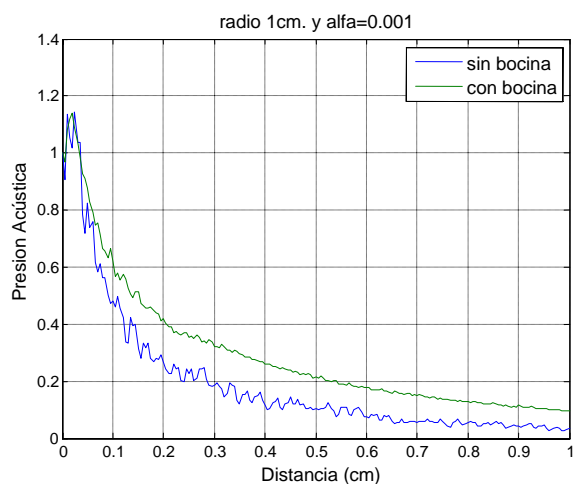


Figura 13. Comparación de simulación con radiación libre y con elemento conformador.

4 RESULTADOS

Si se realiza una comparación entre las simulaciones hechas en radiación libre para varios coeficientes de atenuación y la medida obtenida con el sensor en las mismas condiciones, se obtiene la Figura 14. Como se puede observar en dicha figura, la simulación para los coeficientes de atenuación pequeños no es muy regular y esto es debido, al igual que se comentó con anterioridad, a las limitaciones de memoria en la simulación.

Si se realiza la misma comparación pero haciendo uso del elemento conformador, se obtiene la Figura 15.

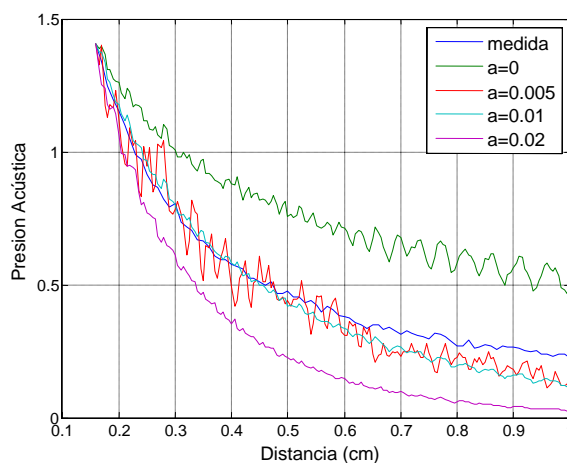


Figura14: Comparación de las simulaciones con distintos coeficientes de atenuación y la medida en radiación libre

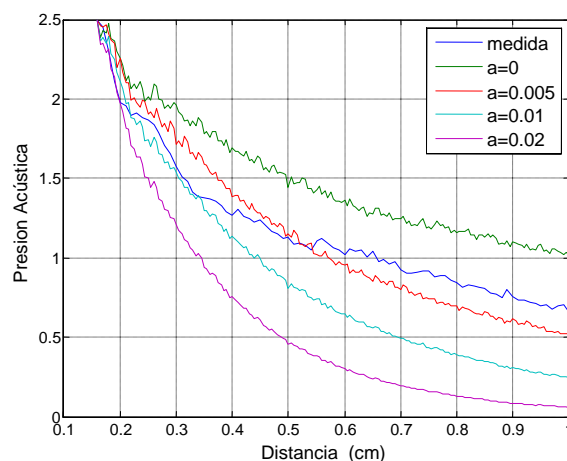


Figura15: Comparación de las simulaciones con distintos coeficientes de atenuación y la medida con bocina.

5 VALIDACIÓN DEL MODELO EF

A la vista de las comparaciones entre las medidas y las simulaciones con distintos coeficientes de atenuación y para el caso de radiación libre y con el uso de bocina, se observa que la validación experimental del modelo simulado con elementos finitos se produce para un coeficiente de atenuación de valor 0.005, valor no muy alejado del valor en las condiciones en las que se realizaron las medidas.

En la Figura 16, se puede observar la relación obtenida entre la presión acústica con elemento conformador y presión acústica en radiación libre, tanto para el caso de la medida como para el caso de la simulación con coeficiente de atenuación de 0.005, y en esta figura se observa que la tendencia de las dos

curvas es la misma. Cabe resaltar de nuevo la limitación hardware que se tiene en la simulación.

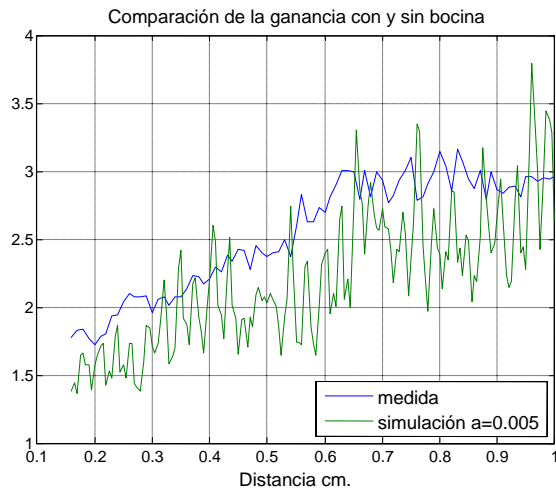


Figura 16: Relación entre presiones acústicas con elemento conformador y en radiación libre para la medida y para una simulación.

6 CONCLUSIONES

En este trabajo se ha encontrado una validación experimental de un modelo simulado para el empleo de elemento conformador.

A partir de los resultados obtenidos con esta validación, el objetivo de trabajos futuros es modificar la geometría y características de la bocina: ángulo de apertura, rugosidad y material con el fin de obtener un modelo por simulación que permita hallar las características óptimas que ha de tener un elemento conformador para conseguir unas características determinadas del lóbulo de radiación para una aplicación predeterminada, especialmente concentrar el haz y alargar la distancia de alcance.

Agradecimientos

Este trabajo ha sido realizado bajo el patrocinio de la CICYT: proyecto DPI2005-08358.

Referencias

- [1] L. Alonso, C. Rodríguez, M. Fernández, S. Robla, E. G. Sarabia, J. Pérez Oria, Conformation, using horns, of radiation beams of ultrasonic sensors. *XXV Jornadas de Automática*. Ciudad Real, España 2004.
- [2] L.Alonso, J.P. Oria, M. Fernández, C. Rodríguez, S. Robla, Qualitative Analysis of the

Influence of Horns on Ultrasonic Lobes, Control and Applications, Cancun 2005.

- [3] J. Arce, J.R. Llata, E.G. Sarabia, J.P. Oria, Automatic Fault Detection Using Ultrasonic Techniques: Expert System vs Signal Processing. *IEEE International Symposium on Industrial Electronics*. Pretoria (South Africa),1998.
- [4] J. Pérez Oria and A.M.Groba, Object Recognition using Ultrasonic Sensors in Robotic Applications. *IECON'93. XIX Annual Conference of the IEEE Industrial Electronics Society*. Hawaii (USA), 1993.
- [5] J.R. Llata, E.G. Sarabia, J.P. Oria, Pattern Recognition with Ultrasonic Sensors: a Neural Network Evaluation. *Sensor Review, Vol 21 N°1*, pp 45-57. 2001.
- [6] J. P. Oria; L.A.Rentería; C.Rodríguez; M..Fernández; J.R.Llata, Ultrasonic Identification Techniques of defective Pieces in Hostile Production Environments. *Forum Acusticum 2002*. Sevilla (España)

Localización topológica basada en visión para robots móviles

Francisco Martín, Vicente Matellán, Pablo Barrera y Jose María Cañas
 Grupo de Robótica, DITTE, Universidad Rey Juan Carlos
 {fmartin,vmo,barrera,jmplaza}@gsyc.es

Resumen

Por localización se entienden las técnicas que se encargan de solucionar el problema de determinar la posición de un robot en su entorno usando sus propios sensores. En este trabajo se presentará una técnica de localización que pretende aportar una nueva manera de localizar un robot equipado con una cámara en un entorno topológicamente estructurado mediante técnicas probabilísticas. Se mostrará su aplicación en entornos tan diferentes como espacios de oficinas o el campo de la Robocup¹, y se comprobará como esta técnica es capaz de adaptarse a las tareas que el robot ha de realizar en estos entornos.

1. Introducción

La localización de un robot en su entorno es uno de los problemas clásicos de la robótica móvil. Es el punto de partida de otros problemas como son la interacción con el entorno, y sobre todo, la navegación. Para la mayor parte de las tareas que un robot móvil tiene que realizar depende de su posición en el mismo. En este trabajo pretendemos dar una solución de este problema para dos tareas muy distintas.

El primer caso es el de la navegación en interiores, concretamente en entorno de oficina. En estos entornos existen una gran cantidad de tareas que un robot puede realizar: llevar el correo de un despacho a otro, teleasistencia o tareas de seguridad. Un robot debe conocer su posición para trazar rutas desde su posición a su destino.

Otro escenario totalmente distinto es la competición de la Robocup. En esta competición, en particular en la categoría de las 4 patas² (fig 1), equipos de robots autónomos compiten entre sí en un juego similar al fútbol. Esta competición lo que pretende es presentar un escenario donde múltiples campos de la robótica pueden ser evaluados entre diferentes grupos de investigación a nivel mundial. En este caso, un robot debe saber su posición en el campo para saber qué hacer cuando



Figura 1: Categoría 4-legged. RoboCup 2006 (Osaka, Japón)



Figura 2: Sony Aibo ERS-7

le llega la pelota, irse a una posición determinada cuando no la tiene, o simplemente, enviársela a un compañero para que éste pueda tomar decisiones más “inteligentes”.

El mecanismo de localización que se propone en este artículo resuelve la localización en entornos tan diferentes como los expuestos anteriormente. Usa un enfoque probabilístico basado en Procesos de Markov Parcialmente Observables (POMDP). Divide el entorno donde el robot se ha de localizar topológicamente atendiendo a las características del entorno y/o las necesidades del robot. Se calcula una densidad de probabilidad del conjunto de estados en los que dividimos el entorno.

La plataforma en la que hemos probado esta técnica es el robot Sony Aibo ERS-7. Este robot tiene aspecto de perro (figura 2), y sus principales características son la locomoción realizada con sus 4 patas, y la sensorización basada en su cámara con resolución de 416x320 píxeles.

El problema de la localización en entornos de oficina ha sido ampliamente estudiado [20], pero la mayor parte de ellos [18] han sido diseñados pa-

¹www.robocup.org

²<http://www.tzi.de/4legged/>

ra robot equipados con ruedas, que cuentan con una información odométrica muy precisa e información sensorial en 360° . Estos métodos no son aplicables a robots con patas. El tipo de información recibida de los sensores de sonar o láser es más fácil de procesar que los de una cámara, que es el principal sensor en nuestro trabajo.

Otra diferencia clave con respecto a los trabajos anteriores en entornos de interior y sobre todo en el campo de la Robocup es su enfoque **topológico**. Un representación topológica del entorno, a diferencia de la representación métrica donde la posición es relativa a unos ejes de coordenadas, divide el entorno en zonas con similares características. Estas zonas no tienen por qué tener las mismas dimensiones. La división topológica es hecha dependiendo de las necesidades derivadas de las tareas de los robots, y mantiene entre cada zona una relación de vecindad. Existen varios trabajos en este sentido, como es [15], pero con robot con ruedas. La idea principal que queremos transmitir es que hay multitud de situaciones en las que una localización tan fina como la métrica no es necesaria, ya que hay multitud de posiciones de un entorno en las que el robot realizaría las mismas acciones con la información de localización. No es tan necesaria tanta precisión.

Técnicas probabilísticas markovianas se han usado en [2], [5] y [14], pero, otra vez, siempre con robots con ruedas y sensores de radar o sonar. En [9] se usa una aproximación markoviana basada en visión, pero la información extraída de las imágenes son histogramas y características invariantes a la escala (SIFT) [13] que son calculadas a partir de un enorme conjunto de imágenes tomadas del entorno. Nuestra aproximación no necesita tomar fotos con anterioridad, solo un mapa 2D del entorno.

En el ámbito de la liga de 4 patas de la Robocup, los métodos de localización usados pueden apreciarse en la tabla 1. La más usada es el método de Monte Carlo [19][6][16][17][21][4]. Este método hace un muestreo en el espacio de estados del robot y mantiene una población de *partículas* que van evolucionando y agrupándose en las posiciones donde cree que el robot se encuentra. En [22] se compara Monte Carlo con el filtro de Kalman [8], que es un filtro adaptativo considera óptimo y que admite no linealidad y de esta manera permite abordar el problema de la localización con estas características. El filtro de Kalman Extendido (EKF) ha sido usado como único método en puntuales trabajos [11][12] en la Robocup, pero aunque el método es potente y se ha probado con éxito en otros ámbitos [3][10], en este caso no es suficientemente flexible para adaptarse a entornos

tan ruidosos. El método difuso [1][7] es una variación del markoviano, pero dividiendo el terreno de juego en un grid donde cada celda almacena una posibilidad difusa tanto de la probabilidad de la celda, como del estado más probable y su incertidumbre. Este último método intenta solucionar el problema del coste computacional de los métodos markovianos puros, pero a diferencia del nuestro, queda limitado por lo estricto de la división del terreno de juego, que en nuestro método puede no ser regular en forma de grid, y que en cada casilla no soporta la posibilidad de ser multimodal. Todos estos métodos al final son un método métrico. La alternativa propuesta es la única que pretende solucionar este problema desde el punto de vista topológico mostrando que es lo suficiente potente como realizar las funciones del robot correctamente.

Método	Nº equipos
Monte Carlo	10
EKF	2
Monte Carlo + EKF	2
Método difuso	1
Triangulación	2

Cuadro 1: Métodos de localización utilizados en la edición de la RoboCup 2004

En la sección 2 mostraremos los fundamentos de la localización markoviana que aplicaremos en la sección 3 para entornos de oficina y en 4 para la competición de la Robocup, donde profundizaremos en sus características particulares. En la sección 5 analizaremos los experimentos llevados a cabo para validar el método. Por último, en la sección 6 presentaremos las conclusiones derivadas de este trabajo.

2. Procesos ocultos de Markov parcialmente observables (POMDPs)

Los POMDPs se caracterizan por modelar procesos estocásticos donde se usa una densidad de probabilidad Bel sobre todos los posibles estados $s \in S = \{s_1, s_2, \dots\}$ en los que el robot puede encontrarse. $Bel_t(S = s)$ representa la creencia de estar en el estado s en el instante t . Al inicio del proceso de localización, dependiendo del conocimiento *a priori* de la posición del robot, $Bel_0(S)$ puede estar uniformemente distribuido

$$Bel_0(s_i) = \frac{1}{|S|} \quad (1)$$

o centrado en el estado donde la posición inicial j se

conoce,

$$Bel_0(s_i) = 0 + \delta, \forall i \neq j, \delta \ll 0,01 \quad (2)$$

$$Bel_0(s_j) = 1 - \delta, \delta \ll 0,01 \quad (3)$$

La actualización de Bel se divide en dos pasos atómicos. El primero es la *fase de movimiento* que se aplica cuando el robot realiza un movimiento o completa una acción. Se utiliza la información de la acción para modificar su creencia sobre su estado. La segunda es la *fase de observación*, donde la creencia se actualiza en función de la información extraída de las entradas sensoriales del robot. Este proceso se realiza cuando se obtiene la información necesaria para cada paso y no tiene porque ser ejecutada de manera alternativa.

2.1. Fase de movimiento

Las acciones de un robot son modeladas como un modelo de acción $p(s'|s, a)$, que es la probabilidad de alcanzar el estado s' si la acción a es ejecutada en el estado s . Para obtener la creencia del conjunto de estados $Bel_t(S')$, se aplica una actualización bayesiana. Con cada acción se aplica:

$$Bel_t(s') = \sum_{s \in S} p(s'|s, a) \cdot Bel_{t-1}(s), \forall s' \in S \quad (4)$$

2.2. Fase de observación

Para calcular la creencia actualizada con la información sensorial $Bel_t(S')$, usamos el modelo de observación $p(o|s')$, que representa la probabilidad de estar en el estado s' habiendo obtenido la observación o .

$$Bel_t(s') = p(o|s') \cdot Bel_{t-1}(s), \forall s, s' \in S \quad (5)$$

3. Aplicación a entornos de interiores

Esta aplicación pretende detectar la posición de un robot móvil en un entorno de oficinas para que pueda navegar por los pasillos de un despacho a otro. Usará como principal sensor las imágenes de la cámara, y utilizará también el sensor de infrarrojos que tiene situado al lado de la cámara en el morro del robot. La información que obtendrá de la imagen de la cámara serán elementos comunes de este entorno, como son las puertas, las luces del techo y los elementos que tiene a ambos lados el robot.

Los elementos que se deben definir en cada entorno de aplicación son:

1. El mapa del entorno y como se traduce a un conjunto de estados.
2. El conjunto de acciones que el robot puede llevar a cabo y el modelo de acción asociado a los estados $p(s'|s, a)$.
3. El conjunto de observaciones que el robot percibe del entorno, y su modelo de observación asociado a los estados $p(o|s)$.

3.1. El conjunto de estados

A todas las posibles posiciones y orientaciones en las que el robot se puede encontrar las denominamos como “estados”. Los estados son definidos sobre un entorno de oficina (figura 3) formado por pasillos (representados como círculos) y habitaciones (representados como cuadrados). No tienen ninguna relación métrica, esto es, un nodo que presente una porción de pasillo con similares características puede tener una longitud mucho mayor que otra porción de pasillo mucho menor. Recordemos que la división es topológica y depende de las observaciones tomadas desde las posiciones del nodo y de las diferentes acciones que un robot puede tomar en una zona del entorno.

En la figura 3 se observa los pasos necesarios para dividir un entorno de oficina. Primero se identifican los nodos como partes del mapa donde el robot percibe idéntica información sensorial y las decisiones adecuadas en cada momento son también idénticas. Como el robot usa la información referentes a los elementos que tiene a los lados (pasillo, puerta o pared), como veremos en la sección 3.3, la división en nodos es realizada teniendo este criterio. De este modo, una porción de pasillo sin puertas a los lados en un mismo nodo, pero si hay una porción de pasillo que tiene una puerta en uno de los dos lados, lo consideramos un nodo diferente.

Finalmente, una vez obtenidos los nodos, como observamos en la figura 3, se obtienen los estados en los que el robot se puede encontrar. Una manera simple y homogénea es dividir cada nodo en cuatro orientaciones, ya que un robot ha de realizar diferentes acciones dentro de un nodo dependiendo si está orientado hacia la pared, una parte del pasillo o la otra.

3.2. Modelo de movimiento

Las acciones que el robot puede tomar en cada posición son simples, avanzar hasta el siguiente nodo, girar 90° hacia la izquierda o girar 90° a la derecha. Estos giros son en lazo cerrado de manera que el robot está obteniendo información de sus sensores, tanto de infrarrojos como, sobre todo,

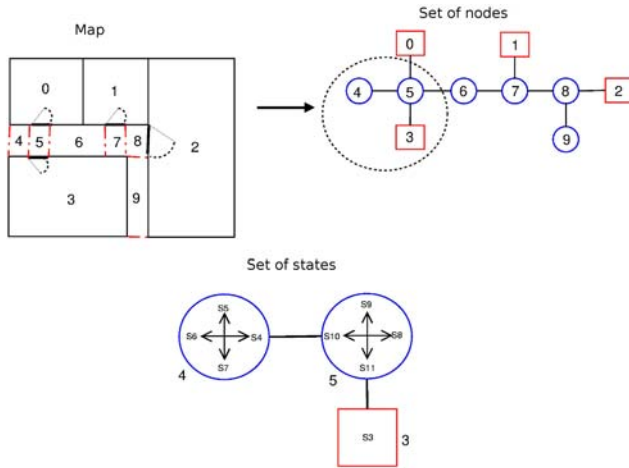


Figura 3: Formación de los estados a partir de la descomposición de los nodos del entorno

de la cámara, para detectar cuando ha cambiado de estado. Por ejemplo, si el robot está orientado hacia el fondo del pasillo y a ambos lados tiene pared, y decide realizar la acción de avanzar, el robot avanzará hacia adelante hasta que en alguno de los lados deje de encontrar pared.

El principio probabilístico de este método nos permite asociar a cada acción la probabilidad de éxito en la realización de la misma. En la tabla 2 queda definida esta probabilidad. Para el caso de los giros a ambos lados, la probabilidad de que el robot no realice el giro correcto y permanezca en el mismo estado (N) es 0.15. La probabilidad de que gire 90° correctamente (G) es 0.70. La probabilidad de que gire hasta el estado situado a 180° (GG) es 0.15, y la probabilidad de que gire hasta el estado situado a 270° (GGG) es cercana a 0. De igual manera, para la acción de avanzar hasta el siguiente estado, la probabilidad de quedarse en el mismo estado (N) es 0.20, de realizar correctamente la acción (A) es 0.6, y de avanzar más allá del estado al que se quiere llegar (AA y AAA) es 0.15 y 0.05 respectivamente.

Esto permite que el sistema sea robusto a colisiones, deslizamientos, errores en la detección del cambio de estado y cualquier eventualidad que pueda suceder en el mundo real donde el robot se ha de desenvolver. Los valores anteriormente mostrados son fruto de una experimentación y observación de la precisión de los movimientos y en la detección de cambios de estado.

El modelo de movimiento $p(s'|s, a)$ queda finalmente definido con el conjunto de estados (que contiene información de vecindad) y la tabla 2. Una matriz $n \times n$, siendo n es número de estados, se genera *a priori* para cada acción. Estas matri-

GIzq	N: 0.15	G: 0.70	GG: 0.15	GGG: 0.0
GDer	N: 0.15	G: 0.70	GG: 0.15	GGG: 0.0
Avan	N: 0.20	A: 0.6	AA: 0.15	AAA: 0.05

Cuadro 2: Incertidumbre en la ejecución de la acción.

ces representan para cada acción a , $p(s'|s)$, y son usadas cuando el robot realiza una acción para el cálculo del *Bel* actualizado.

3.3. Modelo de observación

Los elementos naturales con los que nos encontramos en este entorno son las paredes, las puertas y las luces del techo. Éstas últimas son útiles ya que pueden dar idea de profundidad en un pasillo. Si se evalúa un estado orientado hacia los fondos del pasillo y se detecta una elevada cantidad de luces, indica que el final del pasillo está lejos, y si se detectan pocas significa que el robot se encuentra cerca del final del pasillo.



Figura 4: Detección de 6 luces y 8 puertas en un estado del pasillo alejada del final del mismo.

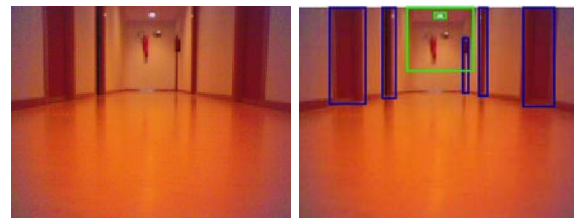


Figura 5: Detección de 1 luz y 5 puertas en un estado del pasillo cercana a la pared.

Las matrices de observación creadas para cada estado en función de la observación que debería tener $p(o|s)$ soportan que haya errores en la extracción de información de las imágenes. Por ejemplo, si un robot en una posición debería detectar 6 luces, no debería agruparse la probabilidad sólo en 6 luces (como en la figura 6), sino que debería tenerse en cuenta error en la observación (figura 6).

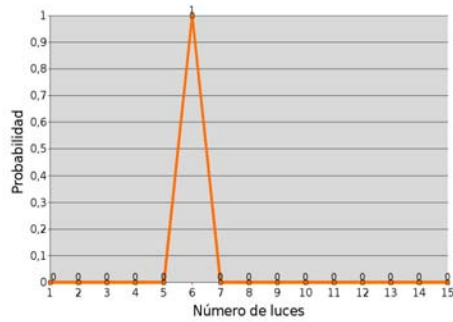


Figura 6: $p(o|s_x)$ sin tener en cuenta errores en la observación si a priori es conocido que $o = 6$ para $s = x$.

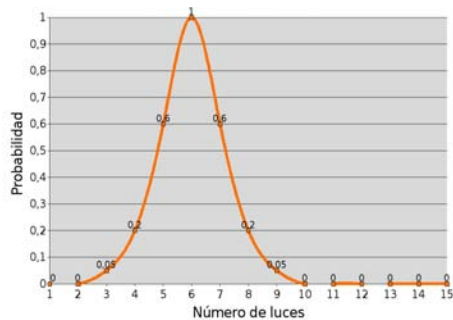


Figura 7: $p(o|s)$ teniendo en cuenta errores en la observación si a priori es conocido que $o = 6$ para $s = x$.

4. Aplicación a la Robocup

El entorno del campo de la Robocup es muy diferente al presentado en la sección 3, y las necesidades de localización son igualmente diferentes. El método propuesto en este artículo es capaz de adaptarse a este entorno, como mostraremos en esta sección.

El campo tiene unas dimensiones fijas de 5400x3600 mm, y tiene una serie de marcas visuales que el robot puede usar para localizarse (figura 8). El objetivo de los robots es el de marcar goles en la portería del equipo contrario. El comportamiento de los robots es diferente dependiendo del sitio del terreno de juego donde se encuentren, y el rol que estén adoptando en cada momento. No solo han de realizar todas sus tareas, incluida la de localizarse, sino que lo han de hacer con unos tiempos de computación razonables para que el robot cumpla con su labor de una manera adecuada.

4.1. El conjunto de estados

Debido a la flexibilidad del método de localización mostrado aquí, la división de entorno del robot en estados se puede hacer de múltiples maneras. Podemos dividirlo en un grid regular donde



Figura 8: Campo de juego de la categoría 4-legged de la RoboCup

todos las celdas tienen las mismas dimensiones y los mismos estados para representar la orientación del robot (figura 9). Esto satisface las necesidades de poder estimar métricamente la posición del robot, que puede necesitar alguna arquitectura de comportamientos en particular. La estimación es semi directa, y el número de estados responde a un compromiso entre la precisión necesitada y los recursos computacionales disponibles.

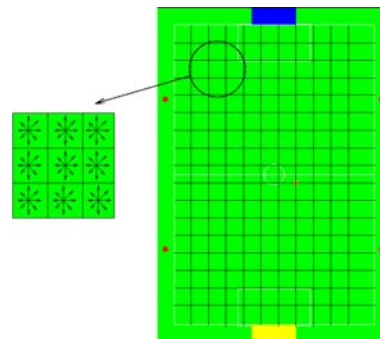


Figura 9: Campo dividido en nodos de la misma dimensión. Cada nodo tiene 8 estados separados de forma uniforme

Otra posibilidad es configurar el espacio de estados dependiendo de las acciones que el robot tenga que hacer en cada zona del campo (figura 10). Por ejemplo, un robot en tareas defensivas en la mitad de su propio campo no necesita saber nada más que si está en la derecha o izquierda del campo, y una orientación aproximada para lanzar la pelota hacia el campo contrario. Esta decisión necesita menos precisión de localización que posiciones cerca de la portería contraria por parte de los delanteros. La localización se orienta hacia el uso que de ella se va a hacer.

4.2. Modelo de movimiento

Al usar una aproximación tipo grid, el modelo de movimiento usado está basado en la odometría. Se realiza un *difuminado* de la probabilidad de cada estado a los cercanos topológicamente, atendiendo a la odometría recibida.

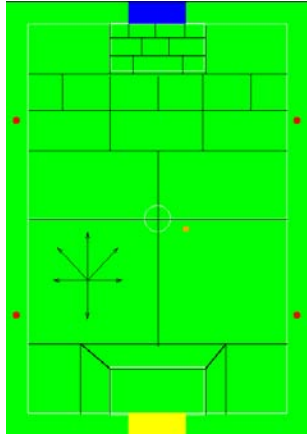


Figura 10: Campo dividido en nodos con distintas dimensiones. Cada uno tiene un número diferente de estados, que corresponden a orientaciones arbitrariamente diseñadas para cada nodo

En la figura 11 se puede ver la evolución del grid de probabilidad atendiendo a la información odométrica. Se observa como se difumina a los estados cercanos, para tener en cuenta errores entre el movimiento real y el percibido por el robot.

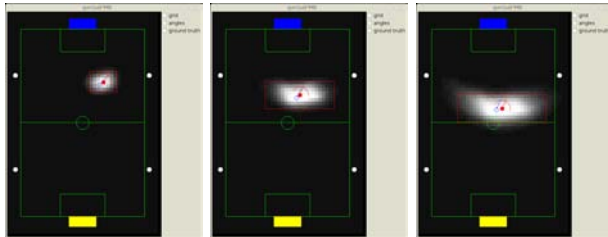


Figura 11: Modelo de movimiento basado en odometría. Se observa la influencia de los estados más probables en sus vecinos.

4.3. Modelo de observación

La información que el robot usa para localizarse es la posición relativa al robot de las marcas visuales que existen en el campo. Estas marcas son las dos porterías y las cuatro balizas. El sistema ha de ser robusto a errores en estas observaciones, ya que el robot está realizando un movimiento continuo, y las medidas pueden no ser exactas.

Internamente el robot almacena tres valores para cada marca visual $\langle \rho, \Theta, anchor \rangle$, donde ρ es la distancia a la marca, Θ es el ángulo respecto al eje del robot, y $anchor$ es un valor que nos indica la confianza de esta información. Si el $anchor$ es elevado, la información se considera correcta y reciente; si es bajo, la información no es válida. Este valor de $anchor$ disminuye con el tiempo desde que la información de la marca visual es refrescada, y con la odometría.

Para hacer el modelo de observación tolerante a errores dentro de nuestro algoritmo, hemos asignado *a priori* la probabilidad de, estando en el estado s , encontrarse con la observación o . En la figura 12 se muestra una observación de una baliza en cierto estado. En ese estado, las probabilidades de encontrar esa baliza están dibujadas en azul. La $p(o|s)$ depende del ángulo con que se percibe la baliza. Si está en la zona oscura es 1, si está en la clara, es 0,6, ya que puede haber habido un ligero error en la observación. Si está en el azul más claro es 0,3, por haberse producido una error considerable, que hay que tener en cuenta. Si está en otra región se considera improbable obtener esa observación en ese estado.

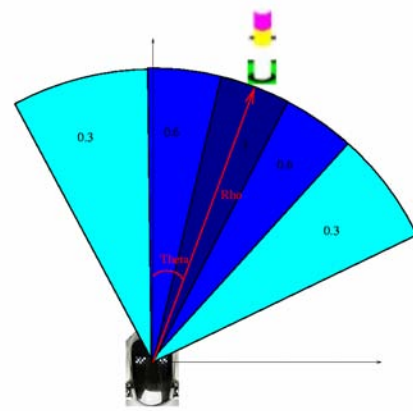


Figura 12: modelo de observación $p(o|s)$.

5. Experimentos

Para medir el método en entorno de interiores, se comparó el estado real del robot con la estimación más probable del robot. En el experimento mostrado en la figura 13 el robot recorre un entorno real de oficinas del punto 1 al 2. Se ha medido la distancia en transiciones de estados (pasos) desde la estimación del robot hasta su posición real, y los resultados se muestran en 14.

Los experimentos sobre el campo de la Robocup consistieron, al no disponer de groundtruth, en colocar al robot en posiciones aleatorias del terreno de juego y hacer que se dirigiera al centro haciendo uso únicamente de la información de localización, y que permaneciese allí. Los resultados después de varias pruebas son que la media del error en X es 297mm, y en Y es 207mm. La desviación estándar de cada uno es 45mm y 130mm respectivamente.

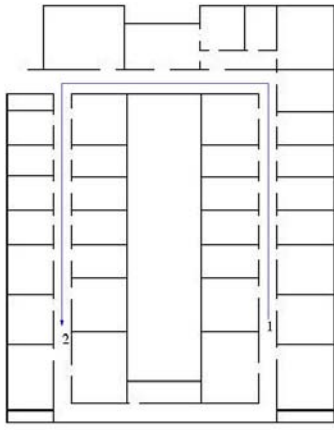


Figura 13: Recorrido del robot en el experimento.

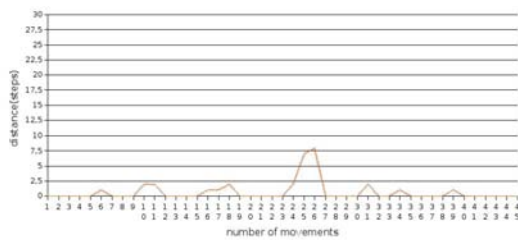


Figura 14: Error en la estimación de la posición.

6. Conclusiones

El enfoque topológico ha resultado ser adecuado en las dos aplicaciones que hemos mostrado en este trabajo. Estas aplicaciones han sido abordadas, sobre todo la segunda, con un enfoque métrico. El enfoque topológico ha permitido almacenar la vecindad entre los estados que en la realidad son adyacentes. Asimismo, ha permitido poder configurar la granularidad de la división del entorno en estos estados, y no siempre esta división puede ser regular. La división puede atender a las acciones que un robot puede tomar en cada parte del juego. Hemos mostrado cómo dividir el entorno en zonas de distinta dimensión donde el robot realizaría acciones idénticas.

Las pruebas realizadas muestran que el enfoque es correcto y funcional en las aplicaciones tomadas en cuenta. Futuras pruebas con groundtruth y un estudio detallado del comportamiento de los robots, en el caso de la Robocup, aportará unos resultados sólidos con los que dar los siguientes pasos. Estos pasos irían encaminados a una posible configuración del espacio de estados dinámicamente en función de los roles de los jugadores, precisión necesitada en cada momento y recursos computacionales del robot.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología, en el proyecto ACRAE: DPI2004-07993-C03-01 y la Comunidad de Madrid en el proyecto RoboCity 2030: S-0505/DPI/0176.

Referencias

- [1] P. Buschka, A. Saffiotti, and Z. Wasik. Fuzzy landmark-based localization for a legged robot. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1205–1210, Takamatsu, Japan, 2000.
- [2] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
- [3] C. Estrada, J.Ñeira, and J.D. Tardós. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.
- [4] Dieter Fox, W. Burgard, H.Kruppa, and Sebastian Thrun. A monte carlo algorithm for multi-robot localization. Technical Report CMU-CS-99-120, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1999.
- [5] Franck Gechter, Vincent Thomas, and François Charpillet. Robot localization by stochastic vision based device. In *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics - SCI 2001 ? The 7th International Conference on Information Systems Analysis and Synthesis - ISAS 2001, Orlando, FL, USA, Jul 2001*.
- [6] Pablo Guerrero and Javier Ruiz del Solar. Auto-localización de un robot móvil aibo mediante el método de monte carlo. *Anales del Instituto de Ingenieros de Chile*, 115(3):91–102, 2003.
- [7] D. Herrero-Pérez, H. Martínez-Barberá, and A. Saffiotti. Fuzzy self-localization using natural features in the four-legged league. In D.Ñardi, M. Riedmiller, and C. Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, LNAI. Springer-Verlag, Berlin, DE, 2004.
- [8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.

- [9] Jana Kosecká and Fayin li. Vision based topological markov localization. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, Barcelona (Spain), April 2004.
- [10] Evgeni Kyriy and Martin Buehler. Three-state extended kalman filter for mobile robot localization. Technical report, Carnegie Mellon, 2002.
- [11] R. Lastra, P. Vallejos, , and J Ruiz-del Solar. Integrated self-localization and ball tracking in the four-legged robot soccer league. In *Proceeding of the 1st IEEE Latin American Robotics Symposium LARS 2004*, México City (México), Oct 2004.
- [12] R. Lastra, P. Vallejos, , and J Ruiz-del Solar. Self-localization and ball tracking for the robocup 4-legged league. In *Proceeding of the 2nd IEEE Latin American Robotics Symposium LARS 2005*, Sao Luis (Brazil), Sept 2005.
- [13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [14] María E. López, Luis Miguel Bergasa, and M.S.Escudero. Visually augmented POMDP for indoor robot navigation. *Applied Informatics*, pages 183–187, 2003.
- [15] Dandapani Radhakrishnan and Illah Nourbakhsh. Topological localization by training a vision-based transition detector. In *Proceedings of IROS 1999*, volume 1, pages 468 – 473, October 1999.
- [16] Thomas Röfer and Matthias Jüngel. Fast and robust edge-based localization in the sony four-legged robot league. In *RoboCup*, pages 262–273, 2003.
- [17] Thomas Röfer and Matthias Jüngel. Vision-based fast and reactive monte-carlo localization. In *ICRA*, pages 856–861, 2003.
- [18] Reid Simmons and Sven Koenig. Probabilistic navigation in partially observable environments. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, pages 1080–1087, Montreal (Canada), July 1995.
- [19] Mohan Sridharan, Gregory Kuhlmann, and Peter Stone. Practical vision-based monte carlo localization on a legged robot. In *IEEE International Conference on Robotics and Automation*, April 2005.
- [20] S.Thrun. Robotic mapping: A survey. *Technical Report CMU-CS-02-111*, 2002.
- [21] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.
- [22] David C.K. Yuen and Bruce A. MacDonald. A comparison between extended kalman filtering and sequential monte carlo technique for simultaneous localisation and map-building. In *Proc. Australian Conference on Robotics and Automation*, Auckland, New Zealand, November 2002.

UN CONTROLADOR BORROSO MULTIVARIABLE EN AGRICULTURA DE PRECISIÓN

Xavier P. Burgos-Artizzu, Ángela Ribeiro
Instituto de Automática Industrial, CSIC
xpburgos@iai.csic.es, angela@iai.csic.es,

Matilde Santos
Facultad de Informática, Universidad Complutense de Madrid
msantos@dacya.ucm.es

Resumen

Partiendo de una plataforma de procesamiento de imágenes, que es capaz de determinar la cantidad de mala hierba presente y el estado de crecimiento de cultivo a partir de una foto, se propone un control borroso MIMO sobre la barra de tratamiento de herbicida de un tractor, que controle la apertura y cierre de cada una de las secciones de la barra, la dosis de herbicida a aplicar en cada instante así como la velocidad del tractor. El controlador se ha desarrollado y simulado en Matlab, con resultados muy satisfactorios que suponen un ahorro significativo en la cantidad de herbicida a utilizar manteniendo un alto nivel de eficiencia en su acción.

Palabras Clave: Control Borroso, Control multivariable, Agricultura de Precisión, Conocimiento experto

b) un notable aumento en el coste económico de la producción agrícola.

La agricultura de precisión nace para conseguir mejorar la calidad de los cultivos, y a la vez ahorrar en su mantenimiento. Uno de sus principales objetivos es rociar fitosanitario únicamente en los puntos estrictamente necesarios y en la cantidad mínima posible, ya que esto permitiría reducir los residuos de herbicida en suelos, plantas y atmósfera, muy dañinos para el planeta. Es decir, en contraposición a los métodos agrícolas tradicionales, propone ajustar las dosis del tratamiento fitosanitario a las necesidades de cada unidad de terreno, aplicando herbicida sólo en aquellas zonas en las que la densidad de mala hierba así lo aconseje, [5]. De este modo se logra reducir tanto la contaminación como el coste de la producción agrícola [3]. Esto implica la recolección de información específica del campo (densidad de cultivo, distribución y cobertura de mala hierba, etc.) en distintos puntos y su posterior procesamiento.

1 INTRODUCCIÓN

Desde que el hombre descubrió la agricultura su principal tarea ha sido siempre separar las espigas de la cizaña. No todo lo que crece en un campo bien cultivado es una bendición, también hay malas hierbas que sólo causan perjuicios a los cultivos, compitiendo con ellos por la luz, el agua y los nutrientes. Actualmente, para eliminar estas malas hierbas, se aplican tratamientos fitosanitarios (herbicidas) en todo el campo a dosis fija. Estas prácticas se orientan hacia una gestión uniforme del campo ignorando la variabilidad espacial que aparece en los cultivos, lo que presenta fundamentalmente dos tipos de inconvenientes:

a) contaminación de los suelos con la consecuente contaminación de las aguas subterráneas. Según informes de la Organización Mundial de la Salud, los niveles de nitrógeno en el agua subterránea han aumentado en muchas partes del mundo como consecuencia de la "intensificación de las prácticas agrícolas",

Aunque la Agricultura de Precisión se marca como objetivo realizar esta tarea en tiempo real, la fase previa, como en toda obra de ingeniería, es la simulación y prueba del sistema. Por ello, los métodos desarrollados hasta ahora se basan en un muestreo de la información ajustado a una malla del campo, que posteriormente se procesa en el simulador y se obtiene como resultado el mapa de aplicación de herbicida, [1], que indica qué cantidad de herbicida aplicaría el sistema en cada punto del campo.

Para llevar a cabo la recolección de información necesaria para el desarrollo del mapa de aplicación de herbicida se puede usar un sistema de visión artificial que, a partir de fotos del campo tomadas en distintos puntos, discrimine la mala hierba del cultivo y calcule tanto la distribución y cobertura de mala hierba como la densidad relativa del cultivo. El uso de sistemas de este tipo permite la detección precisa y puntual de malas hierbas, ya que ajustan sus

valores en función de lo mostrado en cada imagen. En nuestro caso, se dispone de una plataforma desarrollada por el grupo investigador del IAI de procesamiento de imágenes que halla, a partir de una foto, el índice de cobertura de malas hierbas presente en ella, así como el estado de crecimiento del cultivo, necesario para determinar la importancia de la infestación y por ende medir cuantitativamente la cantidad de herbicida a usar [2].

A partir de los valores de índice de cobertura y estado de crecimiento hallados por el sistema de visión artificial, hay que actuar sobre la barra de tratamiento de herbicida del tractor de modo que la cantidad de herbicida suministrada se ajuste al mínimo necesario, entendiendo por mínimo necesario la cantidad de herbicida requerida para eliminar la mala hierba por completo. Para lograr esto, se propone un controlador sobre la barra de modo que ajuste el caudal de herbicida como respuesta a la salida del sistema de visión, y también controle la velocidad del tractor, ya que está influye en la cantidad y distribución de herbicida rociado.

El sistema diseñado ha sido simulado en el entorno de desarrollo Matlab.

En la siguiente sección se describe formalmente el problema, en términos de las entradas y salidas del controlador así como las características y requisitos de la barra de tratamiento. En la sección 3 se analiza en detalle el controlador borroso desarrollado, explicando su estructura y funcionamiento. En la sección 4 se examinan los resultados en comparación con las prácticas agrícolas habituales. Para finalizar, en la sección 5 se exponen las conclusiones más importantes de este trabajo.

2 DESCRIPCIÓN DE LOS COMPONENTES DEL SISTEMA

2.1 TRACTOR Y BARRA DE TRATAMIENTO DE HERBICIDA

El tractor usado es un tractor de la casa John Deer. Es importante destacar que el motor de este tractor (igual que el de todos los tractores modernos), funciona a régimen constante de revoluciones por minuto, denominado toma de fuerza, y que para variar la velocidad de éste no se dispone de un actuador continuo (acelerador), como en los coches, si no de un variador de velocidad discreto, que sólo permite unos determinados valores. Este dato es fundamental a la hora de diseñar el controlador, ya que implica que no todos los valores de salida estarán permitidos.

La barra de tratamiento de herbicida usada es una barra *HARDI NK*, de 10 metros de longitud, con 5

secciones independientes de 2 metros cada una. La barra consta de un depósito de 400 litros, de una bomba que lleva el herbicida hasta los brazos de pulverización, así como de sensores que miden el estado actual (presión de trabajo, caudal actual, secciones abiertas, etc.). El hecho de que la barra disponga de 5 secciones independientes permite mejorar aún más el tratamiento selectivo de malas hierbas, ya que en el mismo instante de tiempo se puede abrir unas secciones y cerrar otras, de modo que se aplique tratamiento en unos lugares y en otros no. Eso sí, el caudal debe ser el mismo para todas las secciones. La limitación de la cantidad de herbicida transportable en el depósito hace necesaria la vuelta atrás para rellenarlo si éste se vacía, de modo que una vez más será preferible usar la mínima cantidad posible de herbicida, para minimizar el número de viajes de recarga del depósito. En la Figura 1 se muestra en detalle una de las secciones de la barra, con su brazo de pulverización.



Figura 1: Detalle de una sección de la barra

2.2 REQUISITOS DEL CONTROLADOR

Como ya se ha dicho, se parte de un muestreo fotográfico del campo, donde cada una de las fotografías tomadas corresponde a una zona concreta, ajustándose a una malla. Posteriormente, se procesa cada una de las imágenes mediante una plataforma de procesamiento de señales bidimensionales que devuelve como salida el valor de *Índice de Cobertura* de mala hierba presente en la foto, así como el *Estado de Crecimiento* actual del cultivo. La primera medida nos da información de la cantidad de mala hierba presente, mientras que la segunda es necesaria para ajustar mejor la cantidad de herbicida ya que ante valores idénticos de índice de cobertura, un campo en estado de crecimiento inicial es potencialmente mucho más peligroso que uno en estado avanzado, debido a que dicho valor de índice de cobertura puede multiplicarse al crecer el cultivo. Además un cultivo recién crecido es mucho más frágil que uno ya bien desarrollado. De este modo, se obtienen dos mapas distintos a partir de las imágenes

iniciales: un mapa de índices de cobertura y otro de estados de crecimiento del cultivo.

A partir de estos dos mapas, hay que lograr que el tractor ajuste la cantidad de herbicida a aplicar en cada instante. Para ello, hay que desarrollar un controlador sobre la barra de tratamiento y el tractor, que reciba como entrada la salida del sistema de visión y el estado actual del tractor, y determine tanto el caudal expulsado (l/min), como la velocidad del tractor (ya que la cantidad final de herbicida rociado es función del caudal y de la velocidad). Además, en nuestro caso, la barra de tratamiento dispone de 5 secciones independientes por lo que el controlador deberá también decidir que secciones cierra y cuáles abre. En la figura 2 se resumen las entradas y salidas resultantes para el controlador. Los valores de salida de velocidad y caudal serán valores de ajuste sobre los valores de entrada (aumentar/disminuir), y dichos valores reajustados se retroalimentan en la siguiente iteración.

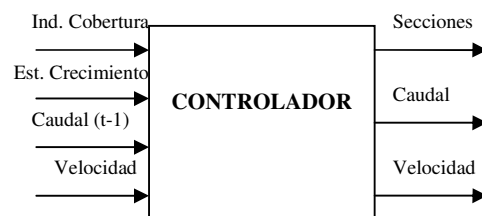


Figura 2: Entradas y salidas del controlador

Un requisito muy deseable (y muy extendido) del controlador es que produzca el menor número posible de cambios en los actuadores que controla, es decir, que produzca el mínimo número posible de variaciones sobre su salida. En nuestro caso, además, resulta más conveniente actuar sobre la velocidad que sobre el caudal, ya que el variador de velocidad es mucho más flexible, preciso y robusto que el regulador de caudal de la barra de tratamiento, expuesto a mucha mayor complejidad e inestabilidad. Siempre que se deba reducir/aumentar la cantidad final de herbicida aplicada, es mejor aumentar/reducir la velocidad, que reducir/aumentar el caudal en la barra. Obviamente, también habrá casos en los que no quedará más opción que actuar sobre el caudal de la barra.

3 CONTROLADOR BORROSO

El controlador debe recibir como entrada las dos variables de los mapas creados por el sistema de visión, el último caudal utilizado y la velocidad actual del tractor, y a partir de estos datos controlar la cantidad y distribución de herbicida actuando sobre

la barra de tratamiento modificando el caudal y abriendo o cerrando las distintas secciones, y sobre el variador de velocidad del tractor para modificar la velocidad. Debido a la independencia de la apertura y cierre de cada sección, y a que cada una cubre un área bastante grande (2 m), el sistema propuesto razona independientemente sobre cada una de las secciones de la barra. Así en cada instante se dispondrá de 5 imágenes distintas, con 5 valores de Índice de Cobertura y Estado de Crecimiento asociados. El controlador generará las salidas para cada una de las secciones, ajustando posteriormente la velocidad al mínimo de las 5 velocidades y el caudal al máximo de los 5 posibles valores para el caudal.

El problema encontrado al diseñar dicho controlador es que no se dispone de ningún modelo del comportamiento que debería seguir, ni tampoco del sistema al que va a controlar. De lo único de lo que se dispone es de conocimiento de experto, es decir, cómo se comportaría un experto agricultor si fuese él el que condujese el tractor y tuviese control sobre la barra de tratamiento. Este tipo de conocimiento es muy difícil de transferir a un controlador clásico, ya que no se basa en datos claros ni precisos, sino en reglas muy intuitivas y con incertidumbre. La Lógica Borrosa, introducida por primera vez en 1965, puede ser de gran utilidad en casos como estos como ya se ha demostrado en numerosas ocasiones [6].

Esta forma de lógica difusa, en contraposición a la Lógica Clásica, permite la representación de valores imprecisos. Las variables en Lógica Borrosa toman valores en un intervalo de confianza de 0 a 1, y posteriormente se combinan mediante reglas borrosas para llegar a un valor final [4]. Esta forma de lógica ha sido muy usada en todo tipo de aplicaciones, especialmente en control. En cuanto a los beneficios a la hora de aplicar Lógica Borrosa en agricultura, un buen ejemplo es [7], donde los autores ajustan las cantidades de herbicida a rociar en campos de cereales de invierno mediante reglas borrosas.

3.1 ESTRUCTURA PRINCIPAL

A la hora de modelar un sistema mediante Lógica Borrosa hay varias configuraciones posibles, tanto para la estructura de las reglas como para la forma en la que se opera o en como se transforma la salida de nuevo a un valor no borroso, o “defuzzyficación”. Esto último es necesario ya que los actuadores necesitan una entrada no borrosa.

En este caso se ha escogido el método de implicación de *Mamdani*, donde tanto los antecedentes como los consecuentes son proposiciones borrosas. Este modelo ha sido preferido a otros por su sencillez, y por el claro hecho de que la salida de nuestro

controlador debe tener forma borrosa. A la hora de seleccionar las funciones matemáticas para las operaciones de AND, OR y “defuzzyficación”, se han seleccionado respectivamente las funciones: MIN, MAX y “Centroide”, que permiten obtener buenos resultados.

3.2 CONJUNTOS BORROSOS

En la figura 2 se mostraban las entradas y salidas del controlador. Todas las entradas toman un valor continuo dentro de un rango especificado, que varía entre una entrada y otra debido a su distinta naturaleza. Dicho valor se debe transformar en un conjunto borroso para su posterior uso como antecedente en las reglas de inferencia del controlador. Para llevar esto a cabo se definen las funciones de pertenencia mostradas en la figura 3. En todas ellas la forma trapezoidal de los conjuntos se ha preferido a otras por su sencillez y por su mayor velocidad de cómputo.

La figura 3A muestran los conjuntos borrosos perteneciente a la variable Índice de Cobertura (IC). El rango de valores, al ser un porcentaje, va desde 0 a 100. La caracterización de IC se ha hecho en 4 conjuntos (nada, poco, medio y mucho). En los extremos (mucho y nada), siempre se aplicará la dosis de herbicida máxima o mínima (nada de herbicida), y se pueden diferenciar claramente cuándo hay poca mala hierba de cuándo hay una cantidad importante. El valor de los rangos ha sido escogido gracias a la experiencia del grupo al tratar con imágenes de cultivos. El Estado de Crecimiento (figura 3B) se ha representado de forma similar con 3 conjuntos (bajo, medio y alto).

La Velocidad (figura 3C) varía entre 0 y 8 (en kilómetros por hora), que son las velocidades habituales del tractor en tareas de laboreo. El caudal (Figura 3D) ha sido normalizado aunque se considera cómo máximo los 400 litros por hectárea de los que es capaz la barra de tratamiento *HARDI NK*.

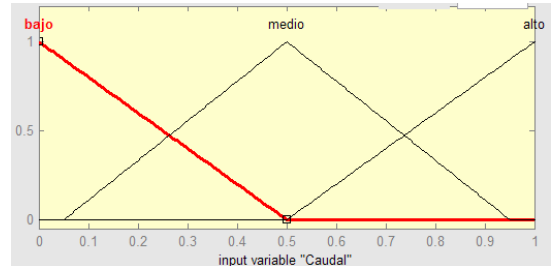
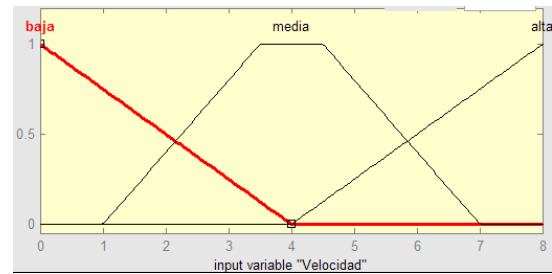
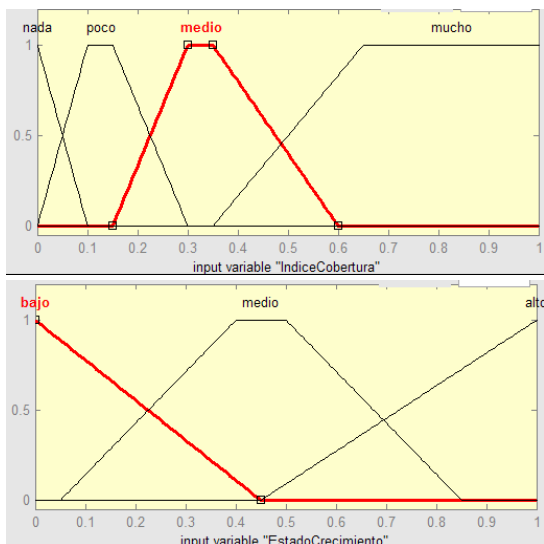


Figura 3 (A, B, C, D, E). Funciones de pertenencia de los conjuntos borrosos de entrada

Del mismo modo, las salidas son conjuntos borrosos definidos por las funciones de pertenencia mostradas en la figura 4, que se usarán para la “defuzzyficación”, dando lugar a las señales de control reales. Aquí la velocidad y el caudal no son valores absolutos, si no el valor de control o modificación que el controlador considera que se debe aportar sobre la velocidad y caudal de entrada. Por ello, la velocidad oscila entre $[-5, 5]$ (5 km/h más rápido o más despacio), y el caudal entre $[-1, 1]$ (bajar del todo o subir del todo). Ambas tienen tres conjuntos representando disminuir, mantener igual (no actuar) y aumentar.

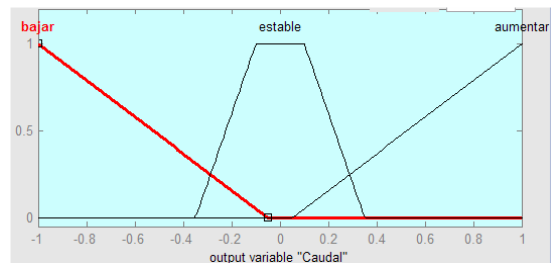
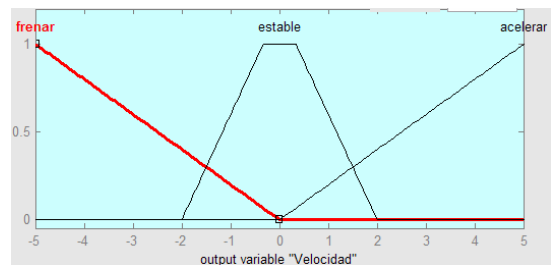


Figura 4: Funciones de pertenencia de los conjuntos borrosos de salida

La otra salida (sección) tiene dos posibles valores (abrir y cerrar sección), por lo que no es necesario ningún conjunto borroso para representarla.

3.3 REGLAS BORROSAS

Una vez convertidas las entradas a conjuntos borrosos, se aplican las reglas de inferencia para hallar las salidas. Para el diseño de las reglas, cómo ya se comentó, ha sido necesario el conocimiento de un experto, que se resume brevemente en la Tabla 1. Las filas representan el valor de entrada (conjunto borroso) de IC, mientras que las columnas representan el valor de entrada del Estado de Crecimiento. El valor de las casillas representa la cantidad de herbicida a aplicar, donde 0 significa no echar nada, 1 echar dosis máxima, $\frac{1}{2}$ dosis media y $\frac{1}{4}$ un cuarto de dosis. Los expertos en la materia consideran suficiente esta separación de las posibles dosis, y no recomiendan añadir mayor complejidad. Cómo ya se explicó al definir los conjuntos de IC, los dos extremos (Nada y Mucho) son aquellos para los que siempre se aplica dosis mínima de herbicida o dosis máxima, dando igual el valor del Estado de Crecimiento. Los otros dos conjuntos (Poco y Medio) sí que dependen del Estado de Crecimiento, ya que cuando el cultivo está poco crecido es potencialmente más peligroso (más frágil ante las malas hierbas), por lo que hay que echar más herbicida.

Tabla 1: Reglas expertas

IC/EC	Poco	Medio	Alto
Nada	0	0	0
Poco	1/2	1/2	1/4
Medio	1	1	1/2
Mucho	1	1	1

Ahora, lo que debe hacer el controlador es ajustar la apertura/cierre de sección, caudal y velocidad para alcanzar una cierta dosis. Dicho ajuste se hace de forma sencilla. Para alcanzar una dosis mínima, basta cerrar la sección, no dejando pasar nada de herbicida. Para el resto de valores, basta abrir la sección y combinar la velocidad con el caudal, teniendo en cuenta que a mayor caudal, más cantidad aplicada. Con la velocidad pasa lo contrario ya que cuánto más rápido va el tractor menos tiempo pasa por encima de la mala hierba y por lo tanto echará menos herbicida. Esto se resume en la Tabla 2.

Tabla 2: Ajuste de la salida

Velocidad	Caudal	Herbicida
Baja	Bajo	1/2
Baja	Medio	1/2
Baja	Alto	1
Media	Bajo	1/4

Media	Medio	1/2
Media	Alto	1/2
Alta	Bajo	1/4
Alta	Medio	1/2
Alta	Alto	1/2

Combinando ambas tablas se forman las reglas borrosas: se miran los valores de IC y EC, para saber que dosis hay que alcanzar (Tabla 1), y con dicho valor se varía el caudal y velocidad de entrada hasta alcanzar el valor deseado (Tabla 2). Siempre se prefiere cambiar la velocidad al caudal, y se procura actuar solo cuando es imprescindible.

4 SIMULACIÓN Y RESULTADOS

Para poder probar el sistema y ver los resultados obtenidos se ha procedido a simularlo en el entorno de desarrollo Matlab.

Se ha considerado un campo de 200 metros de largo por 80 de ancho (4 idas y vueltas del tractor), con una separación entre líneas de cultivo de 1.5 metros. Los mapas de Índice de Cobertura y de Estado de Crecimiento se auto generan aleatoriamente (aunque siguiendo una restricciones que aseguran que lo generado aleatoriamente representa una situación real), para de este modo comparar distintas situaciones. Todos estos valores son ajustables por parámetro y se pueden cambiar entre ejecuciones. Esto se ha hecho con objeto de disponer de una simulación lo más realista y versátil posible.

La figura 5 muestra un ejemplo de ejecución. Las dos primeras imágenes (A y B) presentan los mapas de Índice de Cobertura y de Estado de Crecimiento generados. Nótese como el IC se acumula por rodales (como en los campos reales), y como el EC asemeja la forma de un cultivo. Las tres figuras restantes (C, D, E) representan la salida. La primera muestra los cambios sufridos por el variador de velocidad a lo largo del tiempo, la segunda los cambios sobre el caudal, y la última el mapa de herbicida obtenido.

Analizando en detalle las figuras C y D, comprobamos que los cambios efectuados sobre la velocidad y el caudal son pocos, y que como queríamos, actúa más veces sobre la velocidad que sobre el caudal (un 11.5% de las veces la velocidad, y un 3.8 % el caudal). También se ve que mientras los cambios en la velocidad son bastante contenidos (nunca más de 1km/h de golpe), el caudal varía mucho más bruscamente (oscila entre +0.8 y -0.8). Esto es debido a que el minimizar la actuación sobre el caudal hace que cuando por fin se actúe sea por urgencia, siendo un cambio drástico.

La última figura es la más importante, ya que muestra el mapa de aplicación de herbicida generado. En él se ve claramente como se ajusta en forma y distribución al mapa de Índice de Cobertura, es decir, que muy probablemente eliminará correctamente toda mala hierba. Además, se puede comprobar que la mayor parte de las veces (77.5 %) que se decide aplicar algo de herbicida, la dosis es media. Esto coincide completamente con los estudios realizados hasta la fecha por los expertos, que apunta hacia que la mayoría de las veces aplicar media dosis es suficiente para acabar con las malas hierbas, logrando los mismos resultados que con la dosis completa pero con un ahorro significativo.

Sobre 20 ejecuciones, comparado con prácticas habituales de aplicación de herbicida (a dosis fija en todo el campo), el sistema produce mapas de herbicida en los que de media se emplea 14,17 veces menos herbicida que con dosis total y 7,08 veces menos que con dosis a mitad. Aunque estas son cantidades sujetas a disminuir al trasladarse a la realidad, debido a la infinidad de variables que no se han tenido en cuenta (viento, resistencia de las plantas...), no dejan de ser cantidades que evidencian el claro éxito de la aplicación de técnicas de agricultura de precisión en contraste con las técnicas clásicas.

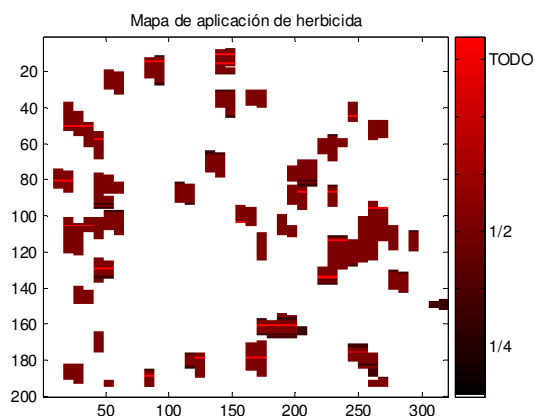
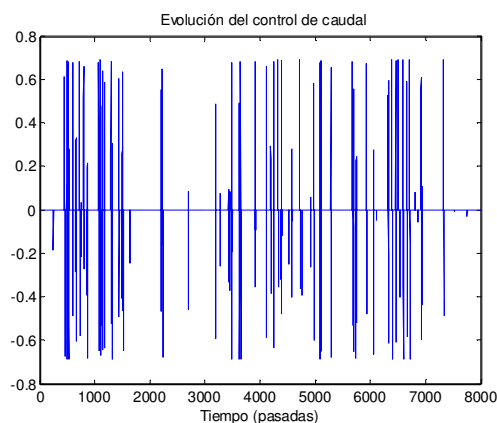
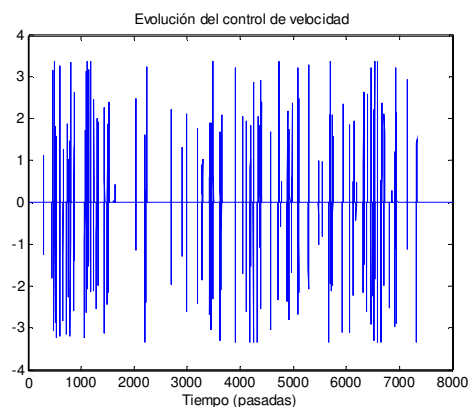
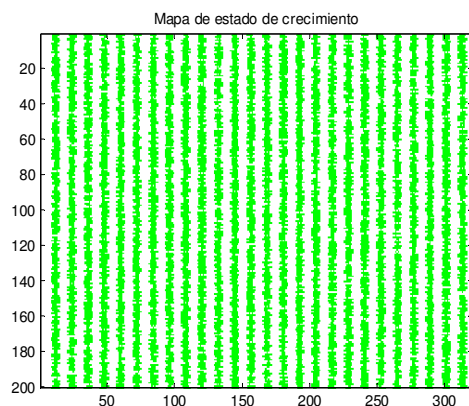
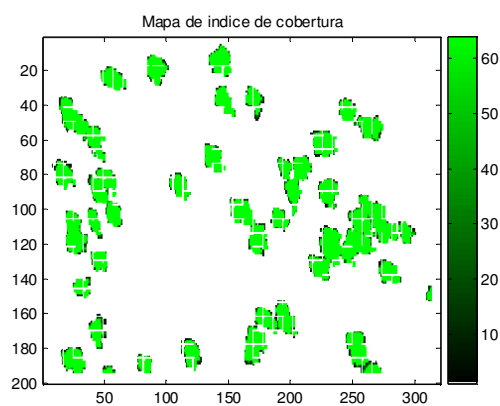


Figura 5 (A, B, C, D, E): Simulación del sistema completo. Mapa de cobertura, estado de crecimiento, acción de control de velocidad, acción de control de caudal y resultado del controlador, respectivamente

5 CONCLUSIONES Y TRABAJO FUTURO

En este artículo se ha presentado un método de control multivariable sobre una barra de aplicación de tratamiento y un tractor comercial agrícola para

dotar a éste de capacidad de tratamiento selectivo de malas hierbas, orientado a una mejor utilización de herbicida, con el consiguiente ahorro económico y alivio para la polución de nuestro planeta, además de facilitar la tarea al agricultor. Además es un controlador que tiene en cuenta múltiples variables a la salida.

El controlador desarrollado usa la salida de un sistema de visión artificial como información para saber dónde y en que cantidad es necesaria la aplicación de herbicida para a continuación actuar directamente sobre los distintos elementos del tractor, minimizando gastos y ejerciendo un control inteligente. Para su diseño e implementación se ha usado la Lógica Borrosa, para tratar la imprecisión de los datos de entrada, así como la incertidumbre sobre la actuación.

Posteriormente se ha simulado dicho controlador, obteniendo resultados muy alentadores, ya que sin pérdida de calidad en los cultivos (se sigue eliminando toda la mala hierba), se consigue ahorrar en el mejor de los casos hasta un 1300% de herbicida.

Como trabajo futuro, se seguirá ampliando el sistema para que incluya variables que no han sido tenidas en cuenta hasta ahora (presión de trabajo, viento, estado de las plantas), y más adelante se incorporará el controlador al tractor.

Agradecimientos

Este trabajo forma parte de un proyecto de investigación financiado por el Ministerio de Educación y Ciencia Extracción de conocimiento espacio-temporal y visión artificial para la automatización de tratamientos localizados (ECOVAAP) CICYT AGL2005-06180-C03-03/AGR.

Referencias

- [1] Cressie N. (1991) "Statistics for spatial data" *New York: John Wiley & Sons, Inc.*
- [2] X.P. Burgos-Artizzu, A. Ribeiro, G. Pajares, A. Tellaeche, J. Barroso, y C. Fernandez-Quintanilla, (2006) "A Computer Vision System for Estimation of Weed Cover" *21st International Symposium on Computer and Information Sciences ISCIS'06.*
- [3] Earl, R., Wheeler, P.N., Blackmore B.S., and Godwin R.J. (1996) "Precision Farming: The management of variability", *In Landwards 51(4) 18–23.*
- [4] García-Alegre, M. C. (1991) "Inteligencia Artificial en el Control de Procesos: controladores borrosos", *Mundo Electrónico, vol. 214, pp. 42-49.*
- [5] Kropff, M.J., Wallinga J., Lotz LAP, (1997) "Modelling for precision weed management". *Precision Agriculture: spatial and temporal variability of environmental quality. Wiley, Chester, 182–204.*
- [6] P.S. Lee, A. L. Dexter (2001): "A Fuzzy Approach to Fault Diagnosis in the Presence of Sensor Bias". *FUZZ-IEEE 2001: 1585-1588*
- [7] Yang, C., Prasher, S.O., Landry, J., Ramaswamy, H. (2003) "Development of an Image Processing System and a Fuzzy Algorithm for Site-Specific Herbicide Applications", *Precision Agriculture, vol. 4, pp. 5-18.*

MODELO DE GESTIÓN ENERGÉTICA INTELIGENTE EN UNA VIVIENDA DOMÓTICA AISLADA

E. Jiménez Macías[†], J.M. Varea Aldama[†], M. Pérez de la Parte[‡], J. C. Sáenz-Diez Muro[†]
Escuela Técnica Superior de Ingeniería Industrial. Universidad de La Rioja.

[†]Departamento de Ingeniería Eléctrica. [‡]Departamento de Ingeniería Mecánica.
e-mail: emilio.jimenez@die.unirioja.es; mercedes.perez@dim.unirioja.es

Resumen

En este trabajo se presenta la implementación en una vivienda aislada de un sistema automatizado de control domótico, no sólo encargado de aumentar las comodidades de los habitantes de la misma, sino incluyendo además un eficiente sistema de gestión energética inteligente, basado fundamentalmente en un modelo del sistema de abastecimiento-consumo, que modela las instalaciones de abastecimiento (energía solar térmica, solar fotovoltaica, y eólica, de manera combinada, a las que se añade una fuente auxiliar no renovable) y la demanda de consumo energético (suministro eléctrico, agua caliente sanitaria, y calefacción).

El dimensionamiento y la estructura de dichas instalaciones se basa en un amplio estudio de necesidades energéticas y posibilidades de generación, para garantizar en la vivienda la demanda energética. De dicho estudio se obtiene el modelo del sistema, que se emplea para la gestión inteligente del sistema energético.

El sistema domótico empleado es el Simon Vis y Simon Vox. Se encarga tanto del control de los elementos habituales de domótica (de confort y comodidad, seguridad y alarmas técnicas, y comunicaciones), como de la gestión energética global del edificio mediante el control coordinado de todas las instalaciones referidas. Dicho control recibe las consignas del sistema de monitorización y simulación, basado en el modelo de funcionamiento energético.

Palabras Clave: Domótica, generación energética, modelo de gestión energética.

1. INTRODUCCIÓN

Si deseamos satisfacer las necesidades energéticas de una vivienda aislada, la forma más factible de hacerlo es empleando energías renovables, tanto desde el punto de vista energético/económico como desde el

ecológico. Además no se debe perder de vista el recientemente aprobado Código Técnico de la Edificación [1], de marzo de 2006, que obliga a la utilización de este tipo de energías en las viviendas nuevas [2].

El trabajo presenta la implementación en una vivienda aislada de un sistema automatizado de control domótico [3], encargado no sólo de aumentar las comodidades de los habitantes de la misma, sino además incluyendo una eficiente gestión energética basada fundamentalmente en el abastecimiento mediante energías renovables.

Para el objetivo de dotar a la vivienda, alejada de las redes de suministro energético, de las mismas disponibilidades energéticas de que puede disfrutar una vivienda que sí tenga acceso a dichas redes, recurrimos a un sistema de abastecimiento basado en energías renovables, en concreto energía solar térmica, solar fotovoltaica, y eólica, a las que se añade una fuente auxiliar no renovable.

El dimensionamiento y la estructura de dichas instalaciones se basa en un amplio estudio de necesidades energéticas y posibilidades de generación, para garantizar en la vivienda tanto el suministro eléctrico como el consumo establecido de agua caliente sanitaria (A.C.S.), y de calefacción.

Con el objetivo de disponer de un sistema automatizado que facilite la habitabilidad de la vivienda, y la dote de mayores niveles de comodidad y seguridad, se emplea el sistema domótico Simon Vis y Simon Vox [3, 4]. Éste se encargará tanto del control de los elementos de confort y las alarmas técnicas, como de la gestión energética global del edificio mediante el control coordinado de todas las instalaciones referidas.

2. INSTALACIÓN DOMÓTICA

Para la realización de la instalación domótica, vamos a desarrollar, una instalación basada en los sistemas domóticos SimonVIS, y SimonVOX. Todo lo que tenga que ver con la automatización propiamente dicha de la vivienda, va a ser gestionada por el

sistema SimonVIS. La parte correspondiente a la seguridad y a las alarmas técnicas, va a ser gestionada por el sistema SimonVOX.

2.1. SISTEMA SIMON VIS

El sistema SimonVIS (Vivienda Inteligente de Simon) es un sistema de control integral de la instalación eléctrica. Este sistema permite la automatización de todos los elementos que conectemos: iluminación, climatización, persianas, toldos, riego, etc. Gracias a sus características técnicas es adecuado para instalaciones de viviendas, locales comerciales, oficinas, escuelas, hospitales, etc.

Para realizar estas funciones dispone de un módulo de control descentralizado, que distribuye las E/S en módulos alejados del módulo de control por medio de un par trenzado de dos hilos, formando una conexión en forma de estrella. Se puede realizar la instalación de forma centralizada o distribuida en cuanto a su conexión.

2.1.1. Topología de la red

Como vemos en el esquema adjunto (Figura 1), su topología de conexión de red es en estrella. Simon Vis se comunica con los módulos de entrada y salida mediante un par trenzado (Figura 2).

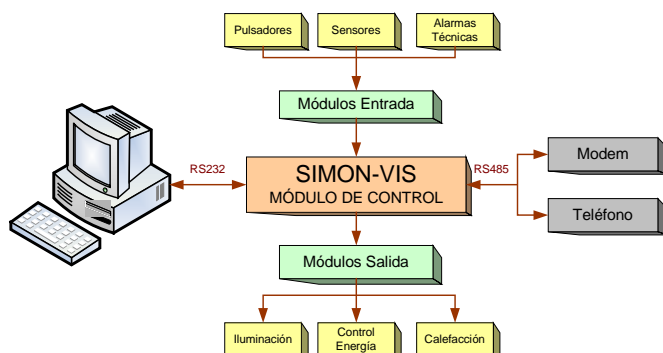


Figura 1: Esquema de conexión en red de Simon VIS

2.1.2. Tecnología de la transmisión

El Módulo de control dispone de 8 Compuertas de Entradas y 16 Compuertas de Salidas para comunicarse con los Módulos de E/S. La comunicación se efectúa a dos hilos, 0 y datos (data).

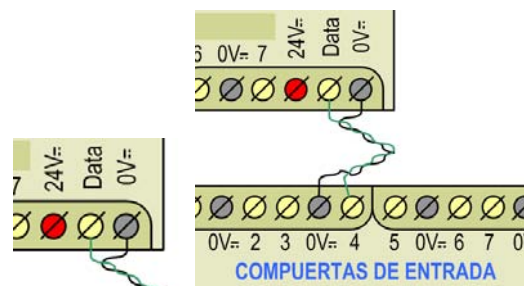


Figura 2: Esquema de conexión de entradas a Simon VIS

El sistema también cuenta con un bus bidireccional normalizado compatible con RS-485. A este bus van conectados los módulos que precisan comunicación bidireccional como el Módulo de Temporizadores y el Módulo de Módem.

2.1.3. Módulos de entradas y salidas

Los módulos empleados tanto de entradas como de salidas son de 230 V. Los podemos ver en los esquemas siguientes (Figuras 3 y 4). Ambos disponen de tres terminales; dos de ellos corresponden a las líneas de 0 y data, para comunicarse con el terminal de control mediante un par trenzado, y el tercer terminal corresponde a 24 V, para aplicar la alimentación positiva a los módulos. Al módulo de entradas le podemos conectar 8 entradas, y al módulo de salidas le podemos conectar 8 salidas.

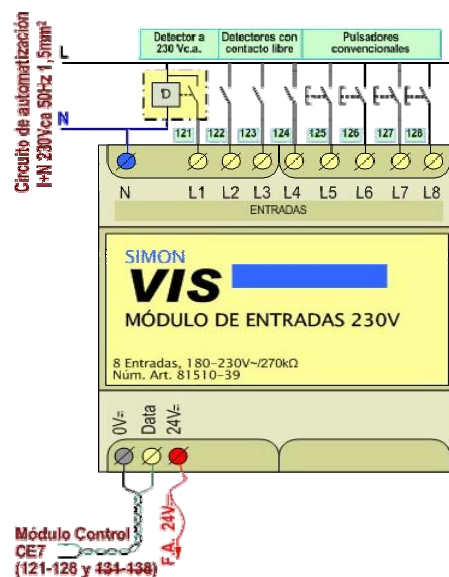


Figura 3: Módulo de entradas de Simon VIS

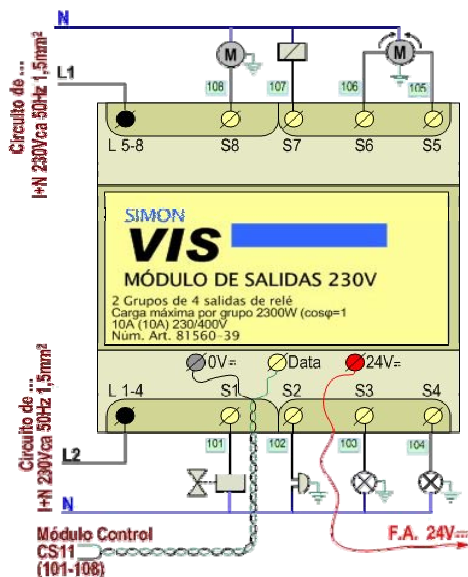


Figura 4: Módulo de salidas de Simon VIS

2.2. SISTEMA SIMON VOX

SimonVOX es una central de telecontrol, ya que está dirigida al control de dispositivos como el de la calefacción y el de aire acondicionado a través del teléfono (Figura 5). Además, SimonVOX efectúa avisos en función de las incidencias producidas dentro del ámbito de la instalación eléctrica en viviendas, pisos, etc. SimonVOX realiza todas sus funciones a través del teléfono, el cual puede ser tanto el propio de la vivienda como cualquier otro exterior. Cabe destacar que un código de acceso garantiza el uso personal del producto.

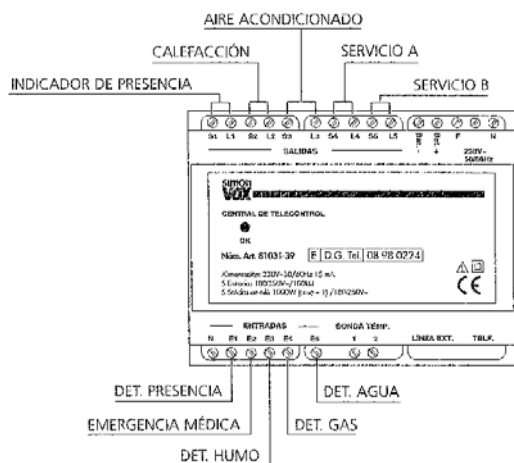


Figura 5: Esquema de conexión de Simon VOX

2.2.1. Entradas y salidas SimonVox

Simon Vox, cuenta con cinco entradas a las cuales conectaremos los detectores adecuados, según el tipo de entrada de que se trate. Las entradas, junto con el

detector que debemos conectar a cada una de ellas, son las siguientes:

Entradas Simon Vox:

- E1: Detector de presencia
- E2: Emergencia médica
- E3: Detector de humo
- E4: Detector de gas
- E5: Detector de agua

Igualmente el módulo de telecontrol Simon Vox cuenta con 5 salidas, a las que conectar un servicio diferente de la vivienda. Dichas salidas, son:

Salidas Simon Vox:

- S1: Indicador del estado del sistema de detección de presencia
- S2: Calefacción
- S3: Aire acondicionado
- S4: Servicio "A" \Rightarrow corte suministro de gas
- S5: Servicio "B" \Rightarrow corte del suministro de agua

Hay que comentar de forma especial las salidas 4 y 5, a través de las cuales Simon Vox permite controlar cualquier carga de la vivienda a través del teléfono, como por ejemplo la lavadora. A esas salidas, se les llama servicio A y servicio B, respectivamente.

También permite Simon Vox particularizar esas salidas, de tal forma que a través de la salida 4, se efectúe el corte del suministro de gas, siempre que sea detectada una fuga de gas en la vivienda, y a través de la salida 5 se efectúe el corte del suministro de agua cuando se detecte una fuga de agua.

2.2.2. Seguridad

Simon Vox efectuará un aviso siempre que se dispare el detector de humo, el detector de gas, el detector de agua, o se produzca una emergencia médica.

Y en función de si el sistema de detección de presencia está activado o no, se efectuará un aviso telefónico ante un disparo del detector de presencia, o un corte del suministro eléctrico (únicamente efectúa avisos por teléfono exterior).

2.2.3. Control del uso del teléfono

Para terminar, decir que Simon Vox permite el control del uso del teléfono. Entre las funciones más importantes, podemos citar:

- Prohibición de cualquier tipo de llamadas
- Restricción del tiempo de duración por llamada
- Programación del sistema de avisos
- Cambiar el número de tonos por llamada a un teléfono exterior
- Cambiar y programar el código de acceso

3. INSTALACIÓN ELÉCTRICA

Para poder realizar la parte del modelo encargada de la producción de energía, inicialmente se ha procedido a analizar los recursos naturales disponibles en la zona.

Se han recopilado datos, sobre la radiación solar y el viento disponible en el emplazamiento en el que está ubicada la vivienda. Una vez analizados todos los datos, se ha llegado a la conclusión, de que la mejor solución va a ser una instalación mixta eólico solar, que garantice el consumo típico de una familia (la consideramos compuesta por un matrimonio y dos hijos).

Las razones por las que la simulación del modelo nos ha proporcionado este sistema como el más rentable es que son sistemas que se complementan perfectamente. Es decir, para satisfacer las necesidades energéticas mediante una sola de las fuentes indicadas, se debe recurrir a una instalación muy sobredimensionada, y que además puede resultar inviable económicamente. Consideremos el caso de la instalación solar fotovoltaica; si queremos satisfacer todas nuestras necesidades energéticas mediante esta fuente de energía, evidentemente deberíamos calcular la instalación de tal forma que no vaya a fallar durante el mes más desfavorable, que es diciembre; eso nos llevaría a una instalación muy sobredimensionada para el resto de los meses del año, especialmente durante los meses de verano.

Como en realidad no podemos controlar los recursos naturales, y nadie nos garantiza que pueda aparecer un período más o menos prolongado en el que no tengamos ni energía solar ni eólica, deberemos disponer además de un sistema de energía de apoyo. En nuestro caso, vamos a emplear un grupo electrógeno para tal fin.

Con arreglo a lo explicado en los párrafos anteriores, nuestro equipo de captación / distribución de la energía, cuenta con los siguientes elementos:

- Aerogenerador.
- Paneles fotovoltaicos.
- Regulador de carga.
- Inversor.
- Banco de baterías.
- Generador de apoyo.

3.1. ESTUDIO DE LAS NECESIDADES A CUBRIR

El primer paso es definir perfectamente los objetivos de la instalación, en cuanto a consumos a satisfacer, atendiendo a las necesidades reales de los futuros usuarios y a sus requerimientos concretos.

Se le va a dar al usuario la opción de efectuar una instalación modular, prevista de forma que resulte fácil ir añadiendo paneles y acumuladores a medida que las necesidades de éste crezcan.

Los elementos constituyentes de la instalación deben guardar entre sí la proporción justa y equilibrada. De nada serviría sobredimensionar el campo de paneles con el propósito de producir más energía si las baterías tienen una escasa capacidad para almacenarla, pues se perdería la mayoría de ella.

3.1.1. Consumo previsto en iluminación

Es preciso elegir las luminarias que más rendimiento proporcionen, descartando las clásicas bombillas de incandescencia, cuya eficiencia luminosa es muy baja, pues aunque resulten más caras son siempre más rentables, al exigir menor potencia eléctrica.

Normalmente se utiliza el alumbrado fluorescente, ya que existe en el mercado una amplia gama de productos con tonalidades y potencias diversas, bien en forma de tubos o de lámparas de reducido tamaño.

Según lo dicho anteriormente, en la Tabla 1, podemos observar la previsión de potencias para alumbrado, junto con los tiempos medios de utilización. Eso nos va a permitir determinar el valor de la energía que debe de ser capaz de suministrar nuestro equipo de generación, para poder cubrir diariamente las necesidades energéticas en alumbrado.

Tabla1: Previsión media de potencias para alumbrado

Estancia	Potencia (W)	Tiempo de utilización (h)	consumo (W·h)
Salón	60	7	420
Cocina	30	7	210
Habitación matrim.	30	3	90
Habitaciones (3)	75	3	225
Baños (2)	60	3	180
Zonas comunes	80	2	160
Garaje	50	2	100
Total			1385

3.1.2. Resto de consumos

De la misma forma que antes se puede estimar el resto de los consumos que vamos a tener en nuestra vivienda (Tabla 2)

Tabla2: Previsión media de otros consumos

Elemento	Cantidad	Potencia	Horas	Consum. (W·h)
TV	1	250	4	1000
Vídeo	1	150	4	600
Hí Fi	1	110	3	330
Frigorífico	1	180	12	2160
Lavadora	1	750	1	750
Pequ. electr.	1	500	4	2000
Total				6840

De las dos tablas anteriores se desprende que de media el consumo total de la vivienda, asciende a 8225 W·h cada día. Con el fin de prevenir posibles sobre consumos, y tiempos de utilización de los aparatos, superior a los indicados, vamos a diseñar nuestra instalación de tal forma que se garantice un consumo superior, en concreto 12.000 W·h al día.

Para satisfacer dichas necesidades energéticas, se van a emplear 10 paneles de 160 W, y un aerogenerador de 1500 W de potencia nominal.

También se ha realizado un pequeño estudio de los consumos energéticos de varias viviendas con características similares partiendo de un modelo no eléctrico, y la media obtenida ha sido de 9935 W·h al día, que es acorde a los datos que nos proporcionaba el otro modelo de consumo.

3.2. DETERMINACIÓN DEL EQUIPO DE ACUMULACIÓN

Lo primero que se ha de determinar, como paso previo al cálculo del acumulador de una instalación, es el número máximo N de días de autonomía previstos para la misma. Dicho número debe ser asignado de acuerdo con las características climáticas de la zona, el servicio que la instalación preste y las circunstancias particulares de cada usuario. En el caso de instalaciones mixtas eólico solares, el número más usual de días de autonomía oscila entre tres y cinco. Con el fin de dotar a nuestra instalación de la máxima fiabilidad, optamos por cinco días.

Además, debemos tener en cuenta el nivel de energía que va a requerir nuestra instalación, y que debe aportar el equipo de acumulación. Recordamos que para cada día, el nivel energético a satisfacer se sitúa en 12000 W·h.

Considerando además que hemos tenido en cuenta otros factores, como son la profundidad de descarga admisible de las baterías, y otras pérdidas que son inherentes a los diferentes elementos que conforman una instalación de este tipo, obtenemos una capacidad del equipo de acumulación de 1889 A·h.

3.3. POTENCIA EÓLICA

3.3.1. Dependencia de la velocidad

La potencia (P) del viento es una función de la densidad de aire (ρ), del área que intercepta el viento (A) y de la velocidad instantánea del mismo (v). El incremento de cualquiera de estos factores aumenta la potencia disponible en el viento:

$$P = \frac{1}{2} \rho A v^3$$

Vemos que la potencia del viento varía con el cubo de su velocidad. Ésa es la razón de poner énfasis en la importancia de instalar la turbina donde los vientos sean mejores.

3.3.2. Variación de la velocidad con la altura

La velocidad del viento, depende de dos factores; de la altura, y de la rugosidad del terreno, es decir, de la orografía. Con el fin de captar mayores velocidades de viento, la turbina eólica irá situada a una altura de 12 metros, por lo que va a ser necesario calcular las velocidades del viento esa altura a partir de los datos disponibles (a 2 metros de altura).

La manera más fácil de calcular el incremento de la velocidad del viento con la altura es el método de la ley exponencial.

$$v = v_0 \left(\frac{H}{H_0} \right)^\alpha$$

Otra aproximación que se usa comúnmente en Europa utiliza la extrapolación logarítmica.

$$v = v_0 \frac{\ln \frac{H}{\alpha}}{\ln \frac{H_0}{\alpha}}$$

En ambos casos:

v = velocidad del viento a conocer.

v0 = velocidad del viento conocida.

H = altura a la que se quiere conocer.

H0 = altura a la que se conoce.

α = coeficiente de rugosidad.

El coeficiente α , es el coeficiente de rugosidad. Depende del tipo de terreno, y en la Tabla 3 podemos ver los valores que toma dicho coeficiente en función del tipo de terreno.

Tabla 3: coeficiente de rugosidad del terreno

Coef. rugosidad	Tipo de paisaje
0,0002	Superficie del agua
0,0024	Terreno completamente abierto con una superficie lisa, (pistas de hormigón en los aeropuertos, césped, etc.)
0,03	Área agrícola abierta sin cercados ni setos y con edificios muy dispersos. Colinas redondeadas.
0,055	Terreno agrícola con algunas casas y setos, de 8 m de altura a una distancia aproximada de 1250 m.
0,1	Terreno agrícola con algunas casas y setos, de 8 m de altura a una distancia aproximada de 500 m.
0,2	Terreno agrícola con algunas casas y setos, de 8 m de altura a una distancia aproximada de 250 m.
0,4	Pueblos, ciudades pequeñas, terreno agrícola, con muchos o altos setos resguardantes, bosques...
0,8	Ciudades más grandes con edificios altos
1,6	Ciudades muy grandes con edificios altos y rascacielos

3.4. ENERGÍA CAPTADA POR EL AEROGENERADOR

Una vez conocidos los recursos eólicos disponibles, se analiza la cantidad de energía que es capaz de captar el aerogenerador, que en este caso ha sido elegido de 1500 W de potencia nominal. Para calcular la energía que capta se ha realizado un estudio exhaustivo de los vientos habidos en la zona, a lo largo del año 2005 (Figura 6).

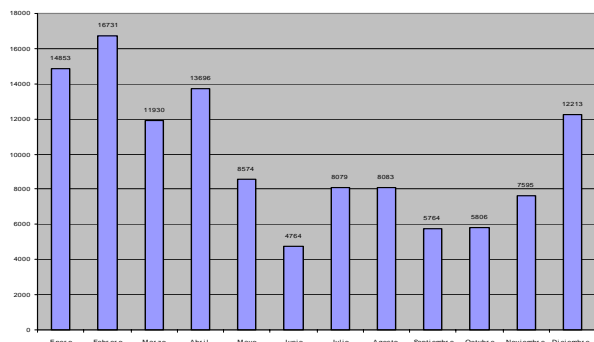


Figura 6: Valores medios de la energía captada por el aerogenerador cada mes, en W·h al día

Puede apreciarse que la mayor captación de energía por parte del aerogenerador se va a producir durante los meses de invierno, ya que durante estos meses ha soplado el viento durante mas horas y a mayor velocidad. Por el contrario, durante los meses de verano, la energía captada por el aerogenerador, desciende bastante. El caso más desfavorable se produce durante el mes de Junio, con una media de 4764 W·h al día.

3.5. ENERGÍA CAPTADA POR LOS PANELES

Se ha analizado el aporte energético de radiación solar a lo largo del año (Figura 7). El aporte mayor se obtiene durante los meses de verano. Concretamente, el mayor aporte energético corresponde al mes de Julio, con una media de 23,3 MJ/m² y día. Por el contrario el mes más desfavorable va a ser Diciembre, con una media de 4,8 MJ/m² y día.

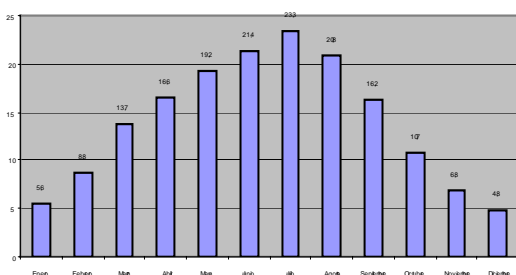


Figura 7: Aporte energético del sol por meses (MJ/m2)

La Figura 8 muestra la energía captada por los diez paneles solares que forman la instalación, para cada mes del año, expresada en W·h al día.

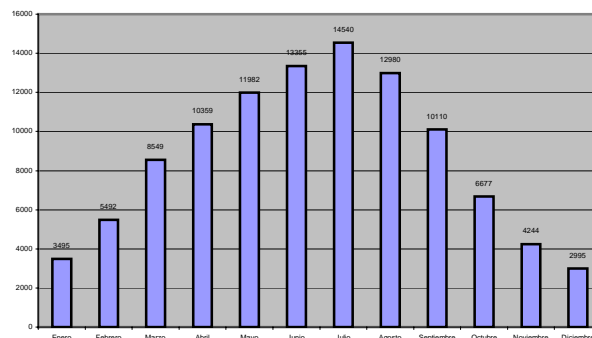


Figura 8: Energía captada por los paneles en W·h al día

3.6. ENERGÍA CAPTADA POR LOS PANELES JUNTO CON EL AEROGENERADOR

El conjunto de paneles más aerogenerador tendrá una producción media diaria, por meses, indicada en la Figura 9. Como podemos observar, el nivel energético suministrado es mucho más estable a lo largo de los meses o de las estaciones.

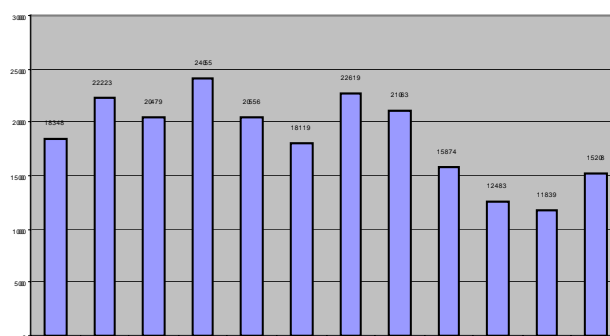


Figura 9: Energía captada por el conjunto de paneles más aerogenerador en W·h al día

4. INSTALACIÓN PARA A.C.S.

Para el consumo doméstico de A.C.S., se suele considerar un nivel que varía entre los 40 y los 70 litros de agua caliente por persona y día. En nuestro país, los consumos rara vez superan los 60 litros.

En cuanto a la temperatura de referencia del A.C.S., suele tomarse siempre igual a 45 °C, si bien en las instalaciones que utilizan energías convencionales para la preparación del A.C.S., ésta se obtiene a veces a una temperatura mucho más alta, lo cual no es necesario en la mayoría de los casos, ni conveniente para la instalación.

La fuente energética que vamos a emplear va a consistir en paneles solares térmicos. Como veremos,

para satisfacer completamente la demanda energética tanto de A.C.S. como de calefacción, vamos a necesitar energía de apoyo. Esa fuente de apoyo va a consistir en una caldera de gas, con quemador para gas propano.

4.1. ENERGÍA NECESARIA PARA A.C.S.

Vamos a analizar la energía necesaria a lo largo del año, para el A.C.S. Para ello, ha sido necesario tener en cuenta una serie de datos, de la vivienda y su entorno, como son el número de ocupantes de la vivienda, su nivel de ocupación, y el consumo estimado por persona y día. Consideraremos una vivienda ocupada por un matrimonio con dos hijos, de edades comprendidas entre los 12 y los 18 años, la vivienda va a estar ocupada durante todo el año, y un consumo medio de 60 litros por persona y día.

También deberemos considerar datos climatológicos, como la temperatura de la zona en la que va a ir ubicada la vivienda, y la temperatura del agua de red, con el fin de poder calcular el salto térmico que es necesario aplicarle al agua, para conseguir que ésta llegue a los puntos de consumo a una temperatura de 45 °C.

Con esos datos se ha realizado un estudio de la energía necesaria, y los resultados obtenidos aparecen en la Figura 10. Como podemos observar el máximo nivel de energía corresponde a los meses de invierno, y el mínimo durante los meses de verano. Eso es lógico, si tenemos en cuenta que el salto térmico que es necesario aplicarle al agua para conseguir que ésta alcance una temperatura de 45 °C, va a ser mayor en invierno que en verano.

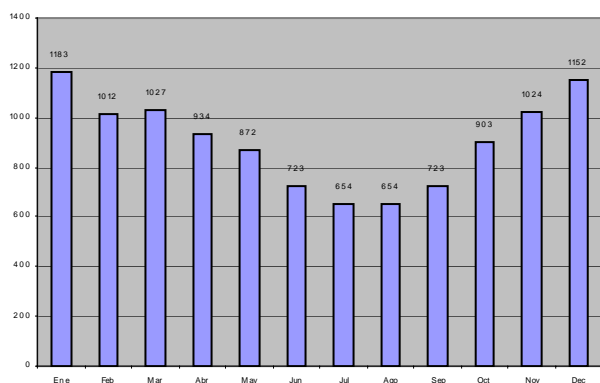


Figura 10: Necesidad energética para A.C.S., por meses, expresada en MJ

4.2. CÁLCULO DE LA SUPERFICIE DE COLECTORES

Una vez conocidas las necesidades energéticas para el A.C.S., vamos a calcular la superficie de colectores necesarios para que aporten esa energía.

Deberemos determinar su inclinación así como la superficie de captación necesaria.

En cuanto a la inclinación, diremos que vamos a favorecer la captación solar en invierno, que va a ser la situación más desfavorable. Con ello, la inclinación de los colectores, va a venir dada por la siguiente regla sencilla: $\alpha = \text{latitud} + 10^\circ$

El siguiente paso, consiste en determinar la superficie de colectores necesaria. El proceso es más largo y laborioso, y es necesario tener en cuenta una serie de datos y consideraciones. Los datos de partida, van a ser los obtenidos en el apartado anterior. Debemos obtener, la energía aprovechable E que incide en un día medio de cada mes sobre cada m² de la superficie inclinada de los colectores; se parte de la tabla general de irradiación horizontal media H para cada mes en la provincia considerada, realizándose si procede una corrección, en función del grado de polución de la atmósfera. Se aplica un segundo factor de corrección por inclinación, k, que depende de la latitud del lugar, y de la inclinación de los paneles.

Hay que tener en cuenta además, una serie de datos como el rendimiento del colector, el número de horas de sol útiles para cada mes del año, las pérdidas a considerar por el desfase entre la captación y el consumo, etc.

Una vez realizados todos los cálculos, hemos obtenido una superficie de colectores de 6 m², lo que supone que debemos colocar 3 colectores solares.

La tabla 4 muestra un balance energético por meses entre la energía necesaria para A.C.S. y la producida por los paneles solares. El déficit lo deberá aportar la fuente de energía auxiliar de apoyo.

Tabla 4: Balance energ. de ACS y paneles térmicos

Mes	Necesidad energética mensual en MJ	Energía producida en MJ	Déficit energético MJ
Ene	1213	128	1085
Feb	1067	515	552
Mar	1120	1309	0
Abr	1023	1377	0
May	1026	1594	0
Jun	963	1860	0
Jul	964	2400	0
Ago	995	2437	0
Sep	993	2071	0
Oct	1057	1364	0
Nov	1083	625	458
Dic	1213	136	1077

Vemos que la mayor parte de los meses del año, no va a ser necesario ningún aporte extra de energía por

parte de la fuente auxiliar. Si que va a ser necesario, aportar energía durante los meses de invierno; especialmente, durante los meses de Enero y de Diciembre. Esos datos se representan en la Figura 11.

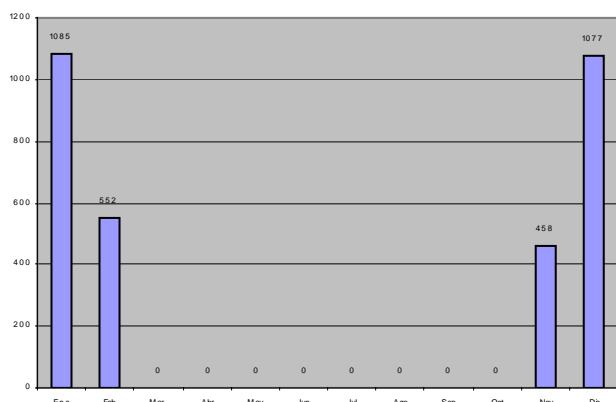


Figura 11: Déficit energético de ACS en MJ

4.3. CÁLCULO DE LA ACUMULACIÓN

Si el volumen de acumulación es demasiado pequeño, la temperatura a la que el calor quedará acumulado será muy elevada. Un correcto dimensionamiento del sistema de acumulación evitará muchos problemas a la instalación. El volumen óptimo de acumulación, depende de tres factores: de la superficie colectora, de la temperatura de utilización del A.C.S., y del tiempo transcurrido entre la captación y el consumo de energía.

4.3.1. Volumen de acumulación, en función de la superficie colectora

La Figura 12 presenta la relación entre el volumen de almacenamiento por metro cuadrado de superficie colectora y el porcentaje de aprovechamiento de la energía captada.

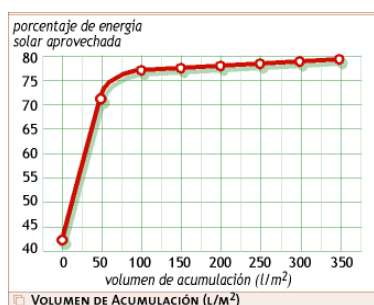


Figura 12: Rendimiento según el volumen de acumulación

Del gráfico se puede desprender que el volumen óptimo de acumulación está comprendido entre los 50 y 70 litros por metro cuadrado de superficie colectora. Para valores menores, el rendimiento es muy bajo, y valores mayores, no aumentan

ostensiblemente el rendimiento pero sí encarecen la instalación.

4.3.2. Volumen de acumulación en función de la temperatura de utilización

Cada instalación es un caso especial, y si debemos trabajar con otros rangos de temperatura, deberemos alejarnos del valor anterior. En la Figura 13 podemos obtener el volumen óptimo de acumulación en función de la temperatura de utilización requerida. Si deseamos reducir el volumen de acumulación debemos obtener el A.C.S. a una temperatura mayor.

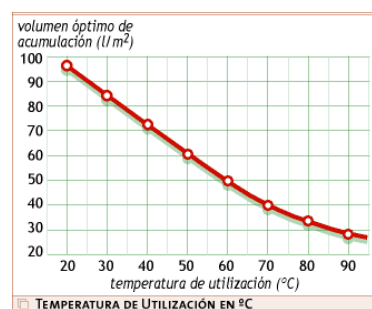


Figura 13: volumen óptimo en función de la temperatura

4.3.3. Volumen de acumulación, en función del tiempo de almacenamiento

Si el periodo de consumo es similar al de captación podemos reducir el volumen de acumulación a valores entre 35 a 55 litros/m². (habitual en procesos industriales). Si el desfase entre captación y consumo es menor de 24 horas, (caso más común en instalaciones de A.C.S.) el volumen recomendado estará entre los 55 y 85 litros/m². Si el desfase se encuentra entre uno y dos días, el volumen recomendado estará entre los 85 a 150 litros/m².

Finalmente, y teniendo en cuenta todas estas consideraciones, ella simulación del modelo ha determinado un depósito acumulador de 300 litros para la vivienda. Este acumulador dispone de dos sistemas para aplicar la energía auxiliar: un serpentín auxiliar de apoyo, al cual vamos a conectar la caldera de gas, y una resistencia eléctrica de 1200 W, a la que le vamos a aplicar los excedentes de energía del sistema eólico solar. Para ello, el regulador de la instalación dispone de una salida, y cuando detecta que el equipo de acumulación se encuentra totalmente cargado, desvía el excedente de energía generada hacia esta resistencia.

5. INSTALACIÓN DE CALEFACCIÓN

Los sistemas convencionales de calefacción utilizan fluido que circula por los radiadores a una

temperatura situada entre los 70 y 90 °C. Estos radiadores ceden calor al aire consiguiendo que la temperatura del local se mantenga constante en un valor deseado. Estas temperaturas de trabajo no pueden alcanzarse con una instalación solar de baja temperatura, ya que a los 80 °C el rendimiento de un colector solar plano es bajísimo. La calefacción por energía solar implica, por tanto, el empleo de otros sistemas de transferencia de calor que funcionen con temperaturas inferiores a los 50 °C. Los sistemas más conocidos son el suelo radiante y los Fan-coils.

En nuestro caso, hemos empleado calefacción por suelo radiante, porque es invisible, proporciona el más alto nivel de confort, mayor seguridad y el menor consumo de energía.

Hemos realizado el cálculo de las cargas térmicas, teniendo en cuenta las pérdidas por transmisión, por infiltración y por renovación. Una vez efectuados todos los cálculos, hemos obtenido unas pérdidas totales en calefacción para el caso más desfavorable, de 8224 Kcal/hora. La fuente energética a emplear, consiste en 6 paneles solares térmicos, como los empleados para la instalación de A.C.S. Además, vamos a necesitar energía de apoyo para los meses mas desfavorables del año, para lo cual haremos uso de nuevo de la caldera con quemador para gas propano.

Referencias

- [1] Código técnico de la edificación (Ministerio de vivienda). <http://www.codigotecnico.org>
- [4] J.C. Sáenz-Díez, J.M. Blanco, E. Jiménez, M. Pérez, JI. Latorre "Optimization of photovoltaic solar electric power for renewable energy generation and DSM strategies in singular apartment buildings", International Conference on Renewable Energies and Power Quality (ICREPQ'06).
- [2] Romero, C. (2005), "Domótica e Inmótica. Viviendas y Edificios Inteligentes", Editorial Ra-ma, 2005.
- [3] Simon S.A., <http://www.simon.es/>

ANÁLISIS DE LA ESTABILIDAD DE UN SISTEMA BORROSO EN TIEMPO DISCRETO CON LMIS

Matías García Rivera, Antonio Barreiro Blas

Departamento de Ingeniería de Sistemas y Automática, Universidad de Vigo,
Campus Universitario As Lagoas-Marcosende 9, 36200, Vigo, Spain
mgrivera@uvigo.es, abarreiro@uvigo.es

Resumen

En este artículo se presenta una técnica de análisis de estabilidad para sistemas de control borroso. Esta técnica es aplicable a sistemas en tiempo discreto y se desarrolla a partir del segundo método de Lyapunov. La condición de estabilidad obtenida puede reescribirse como un problema de factibilidad de un conjunto de LMIs. Los resultados son aplicados a un ejemplo.

Palabras Clave: Control fuzzy, Estabilidad de Lyapunov, Linear matrix inequalities.

1 Introducción

Los sistemas controlados por computadora aparecen en la década de los 50 [1]. En esos años el potencial de las computadoras digitales en el control de sistemas era muy limitado. Estas primeras aplicaciones de las computadoras en el control de sistemas se limitaban al modo supervisor, bien como una guía para el operador o bien como control del punto de trabajo. Por lo tanto, todavía equipos de control analógicos eran necesarios.

Durante la década de los 90, el desarrollo del microprocesador ha tenido un gran impacto en la forma en que las computadoras han sido aplicadas al control de plantas de producción completas. De esta manera cobra especial interés las aplicaciones de control en tiempo discreto, donde los sensores de forma periódica muestrean la salida del sistema, con la cual se genera una actuación para los actuadores de la planta.

La estabilidad de un sistema dinámico es un requerimiento básico para su utilización práctica. La inestabilidad significa que alguna variable del sistema se incrementa más allá de todos los límites y no puede ser

influenciada desde el exterior. Desde un punto de vista de sistemas de control, esto quiere decir que alguna salida del sistema tiende a infinito sin que la entrada pueda hacer nada. Además, el objetivo de un controlador es tanto mantener constante la variable controlada frente a perturbaciones externas (ruido, errores de los componentes, ...), como seguir los cambios prescritos en dicha variable lo más rápido y fielmente posible, y sólo puede lograrse si el lazo de control es estable.

La teoría de Lyapunov permite abordar la estabilidad de un sistema en tiempo continuo o tiempo discreto. La teoría de estabilidad de Lyapunov basada en la estabilidad interna o asintótica del vector de estados, es ampliamente utilizada en el estudio de sistemas no lineales y en sistemas variantes en el tiempo. Las desigualdades matriciales lineales o LMIs pueden expresar el problema de estabilidad definido por Lyapunov.

En la sección 2 se detallará brevemente la teoría de estabilidad de Lyapunov, para después en la sección 3 aplicarla a un sistema no lineal borroso. La sección 4 propone una condición equivalente de estabilidad pero expresada en forma de LMI. Un ejemplo ilustra los resultados obtenidos con esta técnica de análisis en la sección 5.

2 Preliminares

El concepto de estabilidad de Lyapunov juega un papel importante en el análisis de la estabilidad de los sistemas de control descritos mediante ecuaciones en el espacio de estado, tanto en ecuaciones diferenciales como en diferencias ordinarias. La estabilidad de Lyapunov está relacionada con el comportamiento de las trayectorias de un sistema cuando su estado inicial se encuentra cerca de un equilibrio. Dentro del organigrama general de las diversas teorías de estabilidad, la de Lyapunov se engloba en aquella zona que trata con las variables de estado del sistema, y por tanto, tiene que ver con la representación interna de los sis-

temas. En [12, 3, 10, 8, 5, 11] se encuentran estudios completos sobre esta teoría.

Volviendo al caso de funciones cuadráticas para sistemas en tiempo discreto, es decir, con planta $f(\mathbf{x}_k)$ y controlador no lineal $H(\mathbf{x}_k)$, mediante el segundo método de Lyapunov supuesta una función cuadrática $V(\mathbf{x}_k) = \mathbf{x}_k' \mathbf{P} \mathbf{x}_k$ con \mathbf{P} real, simétrica y definida positiva, $V(\mathbf{x}_k)$ será una función de Lyapunov para el sistema considerado si $V(\mathbf{x}_k) > 0$, que ya lo cumple, y si $V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) < 0$, desarrollando

$$\begin{aligned}
 & V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) \\
 &= \mathbf{x}_{k+1}' \mathbf{P} \mathbf{x}_{k+1} - \mathbf{x}_k' \mathbf{P} \mathbf{x}_k \\
 &= [\Phi \mathbf{x}_k + \Gamma K H(\mathbf{x}_k)]' \mathbf{P} [\Phi \mathbf{x}_k + \Gamma K H(\mathbf{x}_k)] - \mathbf{x}_k' \mathbf{P} \mathbf{x}_k \\
 &= \mathbf{x}_k' \Phi' \mathbf{P} \Phi \mathbf{x}_k + \mathbf{x}_k' \Phi' \mathbf{P} \Gamma K H(\mathbf{x}_k) + \Gamma' K H(\mathbf{x}_k) \mathbf{P} \Phi \mathbf{x}_k \\
 &\quad + \Gamma' K H(\mathbf{x}_k) \mathbf{P} \Gamma K H(\mathbf{x}_k) - \mathbf{x}_k' \mathbf{P} \mathbf{x}_k \\
 &= \mathbf{x}_k' \Phi' \mathbf{P} \Phi \mathbf{x}_k + 2\Gamma' \mathbf{P} \Phi \mathbf{x}_k K H(\mathbf{x}_k) + \\
 &\quad \Gamma' \mathbf{P} \Gamma K^2 H^2(\mathbf{x}_k) - \mathbf{x}_k' \mathbf{P} \mathbf{x}_k < 0, \quad \forall \mathbf{x}_k \neq 0 \quad (1)
 \end{aligned}$$

Una desigualdad lineal matricial, también conocida como LMI (*linear matrix inequality*), es una expresión de la forma [2, 6, 7, 9]

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2)$$

donde

- $x = (x_1, \dots, x_m) \in \mathbf{R}^m$ es la variable desconocida.
- F_0, \dots, F_m son un conjunto de matrices reales simétricas conocidas, es decir, $F_i = F_i' \in \mathbf{R}^{n \times n}$, $i = 0, \dots, m$ para un $n \in \mathbf{Z}_+$.
- la desigualdad > 0 indica que la expresión $F(x)$ es definida positiva, por lo tanto, $u' F(x) u > 0$ para todo $u \in \mathbf{R}^n$ y $u \neq 0$.

Muchos problemas de identificación de sistemas y análisis y diseño de controladores, pueden ser formulados o reformulados como LMIs. Esto sólo tiene sentido práctico si una LMI puede ser resuelta eficientemente.

El sistema sobre el que se ha desarrollado y probado este método de análisis, ha sido el modelo linealizado de un servomotor. La dinámica del servomotor se rige por la siguiente expresión:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.7 \end{bmatrix} u \quad (3)$$

donde x_1 y x_2 son respectivamente la posición y la velocidad del servomotor. Las funciones de pertenencia de los antecedentes y de los consecuentes del controlador de lógica borrosa son las indicadas en la figura 1,

y la base de reglas se representa en estructura tabular en la tabla 1. Se verifica que el origen es un punto de equilibrio, y que la característica no lineal del controlador, $H(\mathbf{x})$, se anula en el origen.

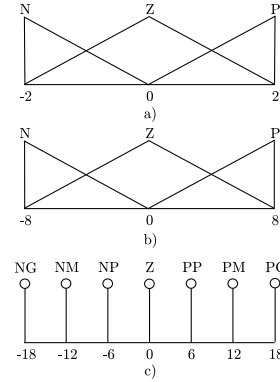


Figura 1: Funciones de pertenencia para el servomotor: a) x_1 ; b) x_2 ; c) u .

3 Estabilidad Borrosa

Como ya se ha visto en la sección anterior, el segundo método de Lyapunov proporciona para sistemas en tiempo discreto con un controlador no lineal, la condición de estabilidad dada por (1). Por lo tanto, el sistema es estable si existe una matriz \mathbf{P} simétrica definida positiva que cumpla

$$\mathbf{x}_k' (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_k + 2\Gamma' \mathbf{P} \Phi \mathbf{x}_k K H(\mathbf{x}_k) + \Gamma' \mathbf{P} \Gamma K^2 H^2(\mathbf{x}_k) < 0, \quad \forall \mathbf{x}_k \neq 0 \quad (4)$$

Esta expresión para el sistema nominal sólo depende de la matriz \mathbf{P} , puesto que el resto de los términos, Φ , Γ , $H(\mathbf{x}_k)$ y K , se encuentran perfectamente definidos.

Dada la dificultad que supone encontrar una \mathbf{P} válida en la superficie de control $H(\mathbf{x}_k)$ para cualquier $\mathbf{x}_k \neq 0$, debido ello a la naturaleza no lineal del controlador de lógica borrosa, simplificaremos el problema buscando dicha matriz en algún volumen que contenga a la superficie de controlador. Los resultados serán más conservadores, es decir, la no existencia de una \mathbf{P} no implica la inestabilidad del sistema, a cambio

$x_1 \backslash x_2$	N	Z	P
N	NP (u_{-1-1})	PM (u_{-10})	PG (u_{-11})
Z	NM (u_{0-1})	Z (u_{00})	PM (u_{01})
P	NG (u_{1-1})	NM (u_{10})	PP (u_{11})

Tabla 1: Base de reglas para el servomotor.

de simplificar la búsqueda de la matriz en el nuevo volumen.

Tal como se demostró en [4], la superficie del controlador de lógica borrosa correspondiente a una celda se encuentra dentro de un tetraedro, cuyos vértices son los puntos de la superficie del controlador evaluados en los cuatro vértices de la celda. Si encontramos una \mathbf{P} válida en el volumen del tetraedro, también será válida en la superficie del controlador correspondiente a esa celda. Según veremos más adelante, para demostrar que una \mathbf{P} es válida en el volumen de un tetraedro, bastará con demostrar que la matriz sea válida en los vértices y aristas de dicho tetraedro.

El espacio de entrada del controlador está dividido en celdas, y en cada una de dichas celdas la superficie del controlador está dentro de un tetraedro. Si la matriz \mathbf{P} es válida en cada uno de estos tetraedros, entonces también será válida en toda la superficie de control. Por tanto, será necesario para garantizar la estabilidad del controlador, demostrar que dicha \mathbf{P} es válida en los vértices y aristas de todos los tetraedros.

La ecuación de Lyapunov para un sistema en tiempo discreto y control borroso dada por (4), para un sistema de orden 2 como es el caso del servomotor, puede reescribirse como la forma cuadrática siguiente

$$L(\mathbf{x}_k, P) = \begin{bmatrix} \mathbf{x}'_k & H(\mathbf{x}_k) \end{bmatrix} \begin{bmatrix} \Phi' P \Phi - P & \Phi' P \Gamma K \\ \Gamma' P \Phi K & \Gamma' P \Gamma K^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ H(\mathbf{x}_k) \end{bmatrix} < 0, \quad \forall \mathbf{x}_k \neq 0 \quad (5)$$

Es interesante realizar algunas consideraciones sobre esta forma cuadrática. Si existe una \mathbf{P} que cumpla la expresión anterior, la superficie $L(\mathbf{x}_k, P) = 0$ es un cono centrado en el origen, puesto que existen puntos donde la función $L(\mathbf{x}_k, P)$ es positiva y también donde es negativa. Observando (5) se deduce que el eje dado por $H(\mathbf{x}_k)$ es siempre positivo, puesto que para $\mathbf{x}_k = 0$ al ser \mathbf{P} definida positiva, entonces $\Gamma' P \Gamma K^2 > 0$ para cualquier Γ y K . Si $L(\mathbf{x}_k, P)$ es de Lyapunov, entonces al menos en los vértices y en las aristas esta función será negativa.

Por ser la superficie dada por $L(\mathbf{x}_k, P) = 0$ un cono centrado en el origen, se pueden dar dos casos: el interior del cono es negativo y no engloba al eje dado por $H(\mathbf{x}_k)$, o el interior del cono es positivo y engloba al eje dado por $H(\mathbf{x}_k)$. Ambas posibilidades pueden verse en las figuras 2 y 3. $L(\mathbf{x}_k, P)$ se puede diagonalizar mediante una transformación ortogonal por ser una forma cuadrática. Si además se aplica un cambio de escala, se puede convertir el sistema de coordenadas original $(x_1, x_2, H(x_1, x_2))$, en otro transformado tal que $\mathbf{z} = (z_1, z_2, z_3)$ y $L(\mathbf{z}, P) = \pm z_1^2 \pm z_2^2 \pm z_3^2$, que cuando uno de los coeficientes posee signo distinto a

los demás, es un cono cuyo eje es la variable que posee dicho signo distinto.

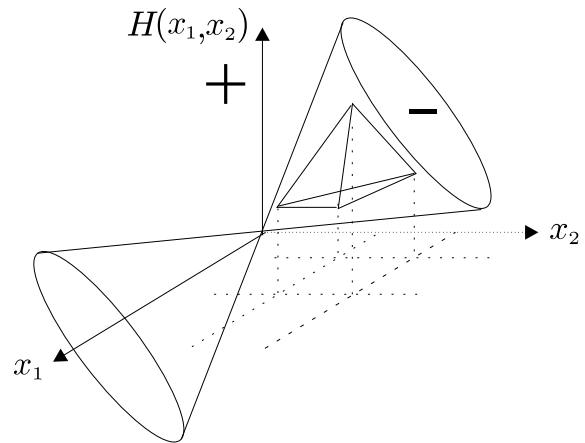


Figura 2: Cono de interior negativo.

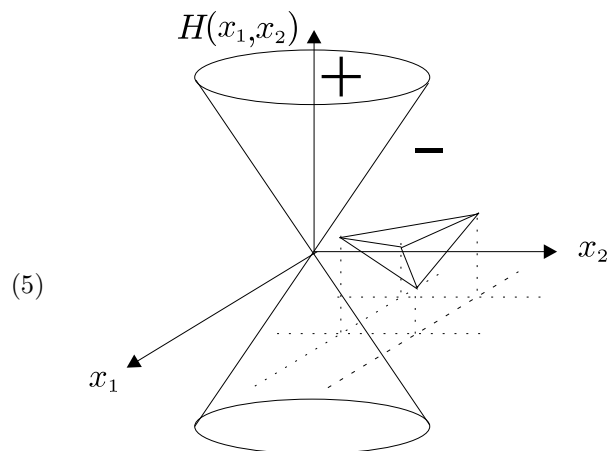


Figura 3: Cono de interior positivo.

Si dos coeficientes del sistema transformado son positivos y uno negativo, el interior del cono transformado es negativo como también lo es el interior del cono del sistema original. En este caso, si la función de Lyapunov evaluada en los vértices del tetraedro es negativa, como el interior del cono es un volumen convexo, todo el tetraedro y la superficie borrosa que encierra también están dentro del cono, y por tanto son negativos. No obstante, este cono de interior negativo debería contener a los tetraedros de todas las celdas, y las celdas se encuentran en todos los cuadrantes, por lo que la única forma válida sería un cono degenerado que abarcara a todo el espacio. Dado que el eje dado por $H(\mathbf{x}_k)$ siempre va a ser positivo, los conos de interior negativo no pueden ser válidos.

Si dos coeficientes del sistema transformado son negativos y uno positivo, el interior del cono transformado

es positivo, como también lo es el interior del cono del sistema original. Si la condición de Lyapunov se cumple en los cuatro vértices de un tetraedro, entonces la función de Lyapunov en ellos es negativa y estos puntos se encuentran en el exterior negativo del cono. Como el exterior del cono no es convexo, el cumplimiento de Lyapunov en los vértices no garantiza que también se cumpla en todo el tetraedro. Será necesaria alguna condición adicional para garantizar que el volumen del tetraedro es siempre negativo.

Sean v_i los vértices de los tetraedros que envuelven la superficie borrosa, donde $i = 0, \dots, n_v - 1$, n_v es el número de vértices y v_0 el vértice situado en el origen. Definimos $L_{v_i}(P)$ como la ecuación de Lyapunov dada por (5) evaluada en el vértice v_i , es decir, $L_{v_i}(P) = L(v_i, P)$. La ecuación de Lyapunov evaluada en los vértices no nulos de los tetraedros correspondientes a todas las celdas debe ser negativa. Por lo tanto,

$$L_{v_i}(P) < 0, \quad i = 1, \dots, n_v - 1 \quad (6)$$

Si existe una P que haga cumplir (5) en los vértices y además en las aristas, tal como muestra la figura 3, entonces sí podemos garantizar que el volumen del tetraedro es siempre negativo. Esto es así porque si en el interior de una cara algún punto fuese negativo, geoméricamente significaría que la cara corta al cono, y además dada la configuración del problema también debería cortar a la arista. Entonces la arista tendría algún punto positivo, lo cual no es posible pues que partimos de que es siempre negativa. De forma similar, si algún punto del volumen fuese negativo, significaría que el cono cortaría el tetraedro por alguna cara.

Sea a_{jk} la arista del tetraedro definida por los vértices $v_j = [\mathbf{x}_{v_j}, H(\mathbf{x}_{v_j})]'$ y $v_k = [\mathbf{x}_{v_k}, H(\mathbf{x}_{v_k})]'$. Esta arista puede expresarse como

$$\begin{aligned} a_{jk}(\alpha) &= \begin{bmatrix} \mathbf{x}_{v_j} \\ H(\mathbf{x}_{v_j}) \end{bmatrix} \alpha + \begin{bmatrix} \mathbf{x}_{v_k} \\ H(\mathbf{x}_{v_k}) \end{bmatrix} (1 - \alpha) \\ &= \begin{bmatrix} \mathbf{x}_{a_{jk}}(\alpha) \\ H_{a_{jk}}(\alpha) \end{bmatrix}, \quad 0 < \alpha < 1 \end{aligned}$$

donde

$$\mathbf{x}_{a_{jk}}(\alpha) = \mathbf{x}_{v_j} \alpha + \mathbf{x}_{v_k} (1 - \alpha), \quad 0 < \alpha < 1 \quad (7)$$

$$H_{a_{jk}}(\alpha) = H(\mathbf{x}_{v_j}) \alpha + H(\mathbf{x}_{v_k}) (1 - \alpha), \quad 0 < \alpha < 1 \quad (8)$$

Definamos $L_{a_{jk}}(P)$ como la ecuación de Lyapunov dada por (5) evaluada en la arista a_{jk} :

$$L_{a_{jk}}(P) = L([\mathbf{x}_{a_{jk}}(\alpha), H_{a_{jk}}(\alpha)]', P), \quad 0 < \alpha < 1$$

La condición de estabilidad de Lyapunov para un control borroso en el caso válido de un cono de interior positivo, es la existencia de una matriz simétrica

definida positiva P tal que, la ecuación de Lyapunov evaluada en los vértices y en las aristas de los tetraedros correspondientes a todas las celdas es negativa, es decir,

$$\begin{aligned} L_{v_i}(P) &< 0, \quad i = 1, \dots, n_v - 1 \\ L_{a_{jk}}(P) &< 0, \quad j = 1, \dots, n_v - 1 \\ &\quad k = j + 1, \dots, n_v - 1 \end{aligned} \quad (9)$$

Como se observa en (9), la condición de estabilidad no incluye aquellas aristas en las que uno de los vértices es v_0 , es decir, el vértice en el origen. Esto es así, puesto que si se cumple la condición de Lyapunov en el vértice no nulo, la recta que une dicho vértice con el origen, jamás cortará al cono centrado en el origen, y por lo tanto, dicha recta siempre tendrá el signo del vértice. Matemáticamente, si $v_k = ([0, 0]', 0)$ entonces las expresiones (7) y (8) se convierten en $\mathbf{x}_{a_{j0}}(\alpha) = \alpha \mathbf{v}_j$ y $H_{a_{j0}}(\alpha) = \alpha v_j$, donde $0 < \alpha < 1$. La ecuación (4) evaluada en la arista a_{j0} es negativa, como se demuestra a continuación

$$\begin{aligned} L_{a_{j0}}(P) &= \alpha^2 \left(v_j' (\Phi' P \Phi - P) v_j + 2 \Gamma' P \Phi v_j K H(v_j) \right. \\ &\quad \left. + \Gamma' P \Gamma K^2 H^2(v_j) \right) < 0, \quad 0 < \alpha < 1 \end{aligned}$$

En la figura 4 pueden verse los vértices y aristas en las que habría que comprobar la condición de Lyapunov dada por (9), para un controlador con cuatro funciones de pertenencia en cada variable de entrada. Ni en el vértice dado por punto de equilibrio, ni en las aristas donde un vértice es el punto de equilibrio, es necesario realizar la comprobación.

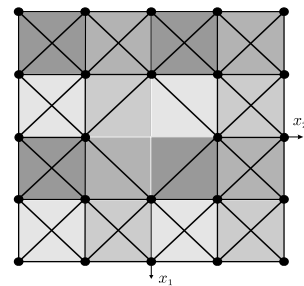


Figura 4: Vértices y aristas donde comprobar condición de Lyapunov.

4 Resolución mediante LMIs

Desarrollando convenientemente la expresión (9) en los vértices y en las aristas, la búsqueda de una P válida puede convertirse en un problema de factibilidad de un sistema de LMIs. Si el conjunto de LMIs que representa al sistema de control borroso es factible, entonces

el controlador será estable y además su solución será la matriz \mathbf{P} buscada.

En un primer paso, la búsqueda de una matriz \mathbf{P} válida en los vértices se transforma en la resolución de un sistema de LMIs. Dados los vértices de los tetraedros a considerar para el estudio de la estabilidad, $v_i = [\mathbf{x}_{v_i}, H(\mathbf{x}_{v_i})]'$, $i = 1, \dots, n_v - 1$, calcular una \mathbf{P} válida en estos vértices es resolver el conjunto de desigualdades dadas por:

$$L_{v_i}(\mathbf{P}) = (\mathbf{x}_{v_i}, H(\mathbf{x}_{v_i}), \mathbf{P}) < 0, \quad i = 1, \dots, n_v - 1$$

es decir

$$\begin{bmatrix} \mathbf{x}'_{v_i} & H(\mathbf{x}_{v_i}) \end{bmatrix} \begin{bmatrix} \Phi' \mathbf{P} \Phi - \mathbf{P} & \Phi' \mathbf{P} \Gamma \mathbf{K} \\ \Gamma' \mathbf{P} \Phi \mathbf{K} & \Gamma' \mathbf{P} \Gamma \mathbf{K}^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{v_i} \\ H(\mathbf{x}_{v_i}) \end{bmatrix} < 0, \quad i = 1, \dots, n_v - 1 \quad (10)$$

Desarrollando según las variables p_{22} y p_{12} de la matriz \mathbf{P} , se obtienen las desigualdades lineales siguientes

$$A_{v_i} p_{22} + B_{v_i} p_{12} < C_{v_i}, \quad i = 1, \dots, n_v - 1 \quad (11)$$

donde A_{v_i} , B_{v_i} y C_{v_i} son números reales que dependen de Φ , Γ y \mathbf{K} . Como se vio en la expresión (??), resolver un conjunto de desigualdades lineales es equivalente a resolver un conjunto de LMIs.

Como último paso se busca una \mathbf{P} válida en los vértices y aristas de los tetraedros. Para ello, se añadirán a las restricciones en forma de LMI dadas por (11) para la búsqueda de una \mathbf{P} válida en los vértices, otras nuevas para que también cumpla Lyapunov en las aristas.

Sustituyendo (7) y (8) en (5), la ecuación de Lyapunov evaluada en la arista a_{jk} es

$$\begin{aligned} L_{a_{jk}}(\alpha) &= L(a_{jk}(\alpha)) = L\left(\begin{bmatrix} \mathbf{x}_{a_{jk}}(\alpha) \\ H_{a_{jk}}(\alpha) \end{bmatrix}\right) = \\ &= \left(\mathbf{x}'_{v_j} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_j} - 2 \mathbf{x}'_{v_j} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_k} \right. \\ &\quad \left. + \mathbf{x}'_{v_k} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_k} \right. \\ &\quad \left. + 2 \Gamma' \mathbf{P} \Phi \mathbf{K} (\mathbf{x}_{v_j} - \mathbf{x}_{v_k}) (H(\mathbf{x}_{v_j}) - H(\mathbf{x}_{v_k})) \right. \\ &\quad \left. + \Gamma' \mathbf{P} \Gamma \mathbf{K}^2 (H(\mathbf{x}_{v_j}) - H(\mathbf{x}_{v_k}))^2 \right) \alpha^2 + \\ &= \left(2 \mathbf{x}'_{v_j} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_k} - 2 \mathbf{x}'_{v_k} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_j} \right. \\ &\quad \left. + 2 \Gamma' \mathbf{P} \Phi \mathbf{K} (\mathbf{x}_{v_j} H(\mathbf{x}_{v_k}) + \mathbf{x}_{v_k} H(\mathbf{x}_{v_j}) \right. \\ &\quad \left. - 2 \mathbf{x}_{v_k} H(\mathbf{x}_{v_k})) \right. \\ &\quad \left. + \Gamma' \mathbf{P} \Gamma \mathbf{K}^2 (2 H(\mathbf{x}_{v_j}) H(\mathbf{x}_{v_k}) - 2 H(\mathbf{x}_{v_k})^2) \right) \alpha + \\ &\quad \mathbf{x}'_{v_k} (\Phi' \mathbf{P} \Phi - \mathbf{P}) \mathbf{x}_{v_k} + 2 \Gamma' \mathbf{P} \Phi \mathbf{K} \mathbf{x}_{v_k} H(\mathbf{x}_{v_k}) \\ &\quad \left. + \Gamma' \mathbf{P} \Gamma \mathbf{K}^2 H(\mathbf{x}_{v_k})^2 \right) = \\ &= A_{a_{jk}}(p_{22}, p_{12}) \alpha^2 + B_{a_{jk}}(p_{22}, p_{12}) \alpha + C_{a_{jk}}(p_{22}, p_{12}) < 0, \\ &\quad 0 \leq \alpha \leq 1 \end{aligned}$$

donde $A_{a_{jk}}$, $B_{a_{jk}}$ y $C_{a_{jk}}$ son funciones lineales en las variables p_{22} y p_{12} , cuyos coeficientes son números reales que dependen de Φ , Γ y \mathbf{K} .

Por lo tanto, la expresión (12) que evalúa la condición de Lyapunov a lo largo de una arista, es una ecuación de segundo grado en la variable α . Para que en una arista se cumpla Lyapunov, es decir, $L_{a_{jk}}(\alpha) < 0, 0 \leq \alpha \leq 1$, partiendo de que sí se verifica en los extremos, $L_{a_{jk}}(0) < 0$ y $L_{a_{jk}}(1) < 0$, se debe cumplir alguna de las siguientes condiciones:

1. $L_{a_{jk}}$ es una parábola convexa o una recta, $A_{a_{jk}} \geq 0$.
2. $L_{a_{jk}}$ es una parábola cóncava, $A_{a_{jk}} < 0$, cuyo máximo se produce para $\alpha < 0$, es decir, $\frac{-B_{a_{jk}}}{2A_{a_{jk}}} < 0$.
3. $L_{a_{jk}}$ es una parábola cóncava, $A_{a_{jk}} < 0$, cuyo máximo se produce para $\alpha > 0$, es decir $\frac{-B_{a_{jk}}}{2A_{a_{jk}}} > 0$.
4. $L_{a_{jk}}$ es una parábola cóncava, $A_{a_{jk}} < 0$, cuyo máximo se produce en el intervalo $0 < \alpha < 1$, es decir, $0 \leq \frac{-B_{a_{jk}}}{2A_{a_{jk}}} \leq 1$, y además es negativo, $\frac{-B_{a_{jk}}^2}{4A_{a_{jk}}} + C_{a_{jk}} < 0$.

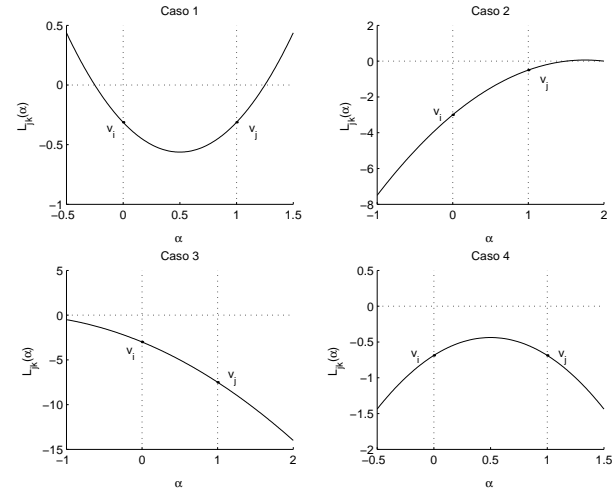


Figura 5: Condiciones para que Lyapunov se cumpla en una arista.

Basta que se cumpla una de las anteriores cuatro condiciones, para que si $L_{a_{jk}}(\alpha)$ es negativa para $\alpha = 0$ y $\alpha = 1$, entonces también lo sea en todo el intervalo $0 \leq \alpha \leq 1$. Estas cuatro posibilidades pueden verse en la figura 5. El caso que se debe descartar es cuando la función de Lyapunov evaluada en la arista es en algún punto positiva, es decir, cuando $L_{a_{jk}}(\alpha)$ es una parábola cóncava y su máximo se encuentra en $0 \leq \frac{-B_{a_{jk}}}{2A_{a_{jk}}} \leq 1$ y además es positivo.

Las condiciones que definen la forma de $L_{a_{jk}}(\alpha)$ son desigualdades lineales en p_{22} y p_{12} , salvo la de la condición de máximo negativo en el punto 4. Esta condición es una desigualdad no lineal, que sin embargo se puede convertir en una LMI mediante las transformaciones siguientes:

$$\begin{aligned} \frac{-B_{a_{jk}}^2}{4A_{a_{jk}}} + C_{a_{jk}} &< 0 \\ -B_{a_{jk}}^2 + 4A_{a_{jk}}C_{a_{jk}} &> 0 \end{aligned}$$

y puesto que en el caso 4, $A_{a_{jk}} < 0$, se puede escribir como la LMI siguiente

$$\begin{bmatrix} 2A_{a_{jk}} & B_{a_{jk}} \\ B_{a_{jk}} & 2C_{a_{jk}} \end{bmatrix} < 0$$

En definitiva, la verificación de la estabilidad de un control borroso se transforma en la búsqueda de una matriz \mathbf{P} , que haga cumplir la condición de Lyapunov en todos los vértices no nulos, y además, en todas las aristas cumpliendo alguna de estas condiciones:

1.

$$A_{a_{jk}} \geq 0 \quad (13)$$

2.

$$\begin{aligned} A_{a_{jk}} &< 0 \\ B_{a_{jk}} &< 0 \end{aligned} \quad (14)$$

3.

$$\begin{aligned} A_{a_{jk}} &< 0 \\ B_{a_{jk}} &> -2A_{a_{jk}} \end{aligned} \quad (15)$$

4.

$$\begin{aligned} A_{a_{jk}} &< 0 \\ B_{a_{jk}} &> 0 \\ B_{a_{jk}} &< -2A_{a_{jk}} \end{aligned} \quad (16)$$

$$\begin{bmatrix} 2A_{a_{jk}} & B_{a_{jk}} \\ B_{a_{jk}} & 2C_{a_{jk}} \end{bmatrix} < 0$$

Por lo tanto, el número máximo de sistemas de LMIs a resolver viene dado en función del número de aristas y será de 4^{n_a} , siendo n_a el número de aristas del controlador. Cada sistema de LMIs estará formado por $n_v - 1$ desigualdades lineales que definen el cumplimiento de Lyapunov en los vértices, más un número variable de LMIs dependiendo de las condiciones en las aristas que se estén considerando.

5 Ejemplo de aplicación

Una vez presentado el método para el análisis de la estabilidad de sistemas en tiempo discreto con control borroso, a continuación se aplican estas ideas al servomotor definido por (3).

Dadas las características de simetría del controlador borroso y la planta, la condición de estabilidad dada por (9), se analizará en los cinco vértices y en las seis aristas que muestra la figura 6. Se necesita por lo tanto, en el peor de los casos, la resolución de $4^6 = 4096$ sistemas de LMIs. Cuando uno de esos sistemas LMI sea factible, el sistema es estable y no es necesario resolver otro más. No obstante, este número será menor en la práctica puesto que de las cuatro condiciones posibles para cada arista, algunas pueden no ser válidas.

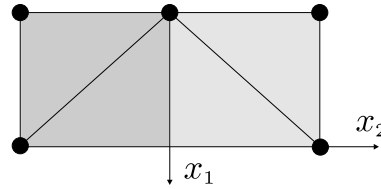


Figura 6: Vértices y aristas donde comprobar Lyapunov para el servomotor.

Se puede simplificar a 864 el número de sistemas LMI a resolver, considerando que en algunas aristas alguna condición no es válida nunca, como se muestra a continuación:

		condiciones				
		1	2	3	4	
aristas	1 :	Sí	Sí	Sí	Sí	4
	2 :	No	Sí	Sí	Sí	3
	3 :	Sí	Sí	Sí	Sí	4
	4 :	No	Sí	Sí	Sí	3
	5 :	No	No	Sí	Sí	2
	6 :	No	Sí	Sí	Sí	3

$$4 \times 3 \times 4 \times 3 \times 2 \times 3 = 864$$

Una matriz válida obtenida es

$$P_{ini} = \begin{bmatrix} 1 & 0.2761 \\ 0.2761 & 0.1111 \end{bmatrix}$$

Para el sistema nominal se puede encontrar una \mathbf{P} de Lyapunov que cumpla la condición dada por la expresión (9), hasta un tiempo de muestreo máximo de $h_{max} = 2.9021$ s. Para este período de muestreo la única combinación de condiciones válidas en las aristas es

arista:	1	2	3	4	5	6
condición:	4	3	4	3	3	2

y la única P encontrada es:

$$P = \begin{bmatrix} 1 & -1.0092 \\ -1.0092 & 1.0897 \end{bmatrix}$$

Obviamente, cuanto mayor es el período de muestreo peor es la respuesta temporal del sistema, por lo que no interesa forzar tiempos de muestreo muy altos pues se perdería calidad en el control.

Referencias

- [1] K. J. Åström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, Upper Saddle River, NY, third edition, 1997.
- [2] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.
- [3] C.A. Desoer and M. Vidyasagar. *Feedback systems: input-output properties*. Academic Press, New York, 1975.
- [4] A. Espada. *Estabilidad y robustez de sistemas no lineales de control. Análisis y diseño de controladores borrosos*. Tesis doctoral, Dpto. Ingeniería de Sistemas y Automática, Univesidad de Vigo, Abril 1996.
- [5] W. M. Haddad and D. S. Bernstein. Explicit construction of quadratic lyapunov functions for the small gain, positivity, circle, and popov theorems and their application to robust stability. part ii: discrete-time theory. *Int. J. Robust Nonlinear Control*, 4:249-265,1994.
- [6] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373-395, 1984.
- [7] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Control Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.
- [8] J-J.E. Slotine and W. Li. *Applied nonlinear control*. Prentice-Hall, Englewood Cliffs, 1991.
- [9] J. V. Tiel. *Convex Analysis: An Introductory Text*. John Wiley and Sons, Chichester, 1984.
- [10] M. Vidyasagar. New directions of research in nonlinear system theory. *Proc. IEEE*, vol 77, pags 1060-1090, 1986.
- [11] M. Vidyasagar. *Nonlinear systems analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [12] J.L. Willems. *Stability theory of dynamical systems. studies in dynamical systems*. Thomas Nelson and Sons Ltd., 1970.

DESARROLLO DE UNA FUNCIÓN DE ENERGÍA PARA PLANIFICAR LA DISTRIBUCIÓN DE RECURSOS HARDWARE EN CHIPS USANDO REDES NEURONALES DE HOPFIELD

F. Javier Sánchez Jurado
Facultad de Informática, UCM,
28040 – Madrid, España
fjsjurado@fdi.ucm.es

Matilde Santos Peña
Facultad de Informática, UCM,
28040 – Madrid, España
msantos@dacya.ucm.es

Resumen

La distribución de recursos dentro de un chip es un problema que, dada su complejidad, es susceptible de ser resuelto aplicando procedimientos de optimización mediante técnicas de inteligencia artificial. En este artículo se propone el uso de redes neuronales recurrentes de Hopfield para la búsqueda de buenas soluciones de forma rápida y eficiente.

La principal dificultad de esta aproximación es formular la función de energía a minimizar de manera que se incluyan todas las restricciones propias del problema.

Los resultados experimentales demuestran que se pueden conseguir buenas soluciones en un tiempo razonable aplicando esta metodología.

Palabras Clave: Redes neuronales, optimización, distribución de recursos, Hopfield.

1 INTRODUCCIÓN

Actualmente, a la hora de diseñar un circuito integrado uno de los aspectos más complicados de solucionar es la distribución de los distintos elementos. Algunos de los parámetros a tener en cuenta son la potencia que disipa cada unidad o la distancia de las conexiones internas a implementar.

Si en lugar de trabajar con un circuito integrado, estamos desarrollando un mapa de bits para una FPGA [4], el problema de la distribución de los elementos es aún más acuciante puesto que la cantidad de recursos disponibles para la construcción de las rutas es limitada.

Teniendo ambos modos de diseño en consideración, se puede concluir que la minimización de la distancia entre los elementos a colocar es un problema relevante a ser resuelto. No sólo por la disminución de las latencias, sino también por la reducción en el consumo relativo a los distintos buses de comunicaciones.

La búsqueda de la mejor solución a este problema es demasiado costosa como para realizarla de forma exhaustiva. En este trabajo se propone un método basado en redes neuronales artificiales para encontrar buenas soluciones en un tiempo relativamente corto.

La aplicación de este tipo de técnicas de inteligencia artificial a la resolución de problemas de optimización no es algo novedoso. Existen propuestas previas para su uso en campos como la mejora del microcódigo de procesadores CISC [2]. El desarrollo de este trabajo toma como modelos esta clase de propuestas para enunciar un procedimiento general para resolver el problema formulado previamente.

A continuación se describe la estructura del artículo. En la sección 2 se plantea el problema de optimización a ser resuelto. A su vez, la sección 3 resume las características principales de la red neuronal que se va a utilizar y en la sección 4 se aplica el algoritmo con las especificaciones propuestas. Los resultados se muestran en la sección 5. Las conclusiones y futuras mejoras finalizan este trabajo.

2 PLANTEAMIENTO GENERAL DEL PROBLEMA

Queremos situar en n huecos libres dentro de un chip a n módulos distintos. Suponemos que la potencia

disipada por cada uno de ellos no es relevante a la hora de elegir la situación. Los módulos se comunican entre sí mediante enlaces, que para simplificar se consideran no dirigidos. El objetivo que se persigue es encontrar una disposición que permita colocar todos los módulos de forma que la longitud media de los enlaces sea lo más corta posible.

La idea consiste en representar la colocación de los módulos mediante una matriz binaria \mathbf{M} , de manera que si el módulo j ocupase la posición i en el chip entonces el elemento m_{ij} toma el valor 1. En otro caso 0.

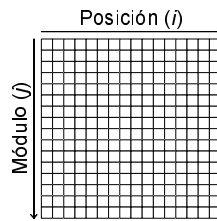


Figura 1: Representación de la matriz \mathbf{M} .

A su vez, el grafo de dependencia de los módulos se expresa mediante la matriz simétrica binaria \mathbf{C} . En ella el elemento c_{ij} vale 1 si el módulo i está relacionado con el j y 0 si no se cumple esta condición.

En cuanto a la distancia entre huecos, está queda reflejada en una matriz simétrica \mathbf{D} tal que el elemento d_{ij} vale la longitud del enlace entre el hueco i y el j .

3 REDES NEURONALES DE HOPFIELD

Las redes de Hopfield fueron propuestas por el físico John J. Hopfield en 1982 [1]. Son un tipo de red neuronal artificial recurrente en la que la capa de entrada es igual a la capa de salida y no hay capas ocultas.

$$U_{oculta} = \emptyset \wedge U_{entrada} = U_{salida} \quad (1)$$

Cada neurona recibe a la entrada todos los valores de las salidas, sin tener en cuenta el suyo propio.

$$C = U \times U - \{(u, u) | u \in U\} \quad (2)$$

Los pesos de las conexiones tienen que ser simétricos.

$$\forall u, v \in U \Rightarrow w_{uv} = w_{vu} \quad (3)$$

En el siguiente gráfico se muestra un ejemplo de red neuronal de Hopfield con 3 nodos en la que no se han especificado pesos, umbrales o estado de activación de la neurona en concreto.

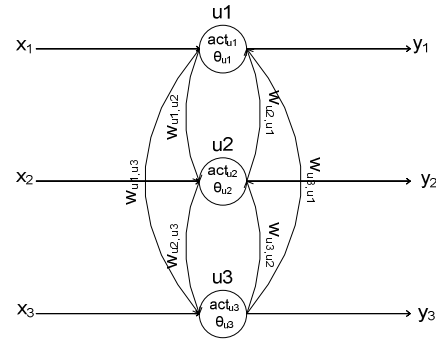


Figura 2: Ejemplo de red neuronal de Hopfield.

De esta forma, el valor de la función de entrada de la red para una neurona se calcula mediante la suma ponderada de todas las salidas.

$$\forall u \in U \quad f_{red}(u) = \vec{w}_u \cdot \vec{f}_{entrada}(u) \quad (4)$$

$$\forall u \in U \quad f_{red}(u) = \sum_{v \in U - \{u\}} w_{uv} \cdot f_{salida}(v) \quad (5)$$

La función de activación se suele definir utilizando un cierto valor umbral para cada neurona.

$$\forall u \in U \quad f_{activación}(u) = \begin{cases} 1, & \text{si } f_{red}(u) > \theta_u \\ act_u, & \text{si } f_{red}(u) = \theta_u \\ 0, & \text{si } f_{red}(u) < \theta_u \end{cases} \quad (6)$$

La salida de las neuronas es la función identidad del estado de activación.

$$\forall u \in U \quad f_{salida}(u) = act_u \quad (7)$$

Conocidos el estado de activación de las neuronas de la red en un instante dado, los valores de los pesos y los distintos umbrales, se define la función de energía mediante una función de Lyapunov [3] de la siguiente forma:

$$E = -\frac{1}{2} \sum_{u,v \in U | u \neq v} w_{uv} \cdot act_u \cdot act_v + \sum_{u \in U} \theta_u \cdot act_u \quad (8)$$

Por otra parte, la transición entre estados se puede realizar de dos formas: por relajación paralela o mediante la actualización secuencial de las neuronas.

El primer mecanismo consiste en cambiar el estado de todas las neuronas de forma paralela en función de la activación actual de las demás. Aunque se trata de una forma muy intuitiva, no se puede garantizar que la red converja a un estado estable. Por lo tanto, la salida depende del momento en que se cancele el proceso recursivo.

En cuanto a la actualización secuencial, es un proceso mucho más sencillo que consiste en calcular sólo el estado de activación de una neurona en cada paso. La neurona a revisar se elige siguiendo un orden preestablecido.

Según el teorema de convergencia para la actualización secuencial [3], toda red de Hopfield tiende a un estado estable en un número finito de pasos independientemente del estado inicial de activación. Dicho estado se corresponde con un mínimo de la función de energía de la red. La salida resultante depende por tanto del estado inicial y del orden de selección de las neuronas. El número de iteraciones necesarias para alcanzar la convergencia es siempre inferior a $n \cdot 2^n$, siendo n el número de neuronas de la red.

Dadas las características de este tipo de redes, su uso se centra en aplicaciones como el diseño de memorias asociativas, reconocimiento de patrones o la resolución de problemas de optimización.

4 ALGORITMO PROPUESTO

Este método se basa en la aplicación del procedimiento general de resolución de problemas de optimización con redes Hopfield. La principal dificultad que se plantea es la correcta formulación de la función de energía, que tiene que representar todas las restricciones necesarias para que una solución se considere correcta.

El primer paso de este procedimiento general consiste en convertir la función que se quiere optimizar en una función a minimizar. Una vez hecho esto hay que transformarla de manera que se corresponda con la forma propia de la función de energía de una red de Hopfield. De esta forma se pueden calcular los valores de las matrices de pesos y umbrales, que son necesarios para construir la red neuronal. Por último hay que realizar sucesivas simulaciones de la red con distintos valores iniciales de activación, tomándose la mejor de todas las soluciones obtenidas como resultado.

Por tanto, utilizando las matrices **M**, **C** y **D** descritas anteriormente, se trata de encontrar los valores de **M** que hacen que la suma de las longitudes de todos los

enlaces sea mínima. Además, no puede haber más que un 1 por fila (9) o columna (10) en dicha matriz.

$$\forall j \in \{1, \dots, n\} \quad \sum_{i=1}^n m_{ij} = 1 \quad (9)$$

$$\forall i \in \{1, \dots, n\} \quad \sum_{j=1}^n m_{ij} = 1 \quad (10)$$

Si tenemos en cuenta que las matrices **C** y **D** son simétricas, la primera restricción se puede expresar mediante la función de energía (11).

$$E_1 = \frac{1}{2} \sum_{j_1=1}^n \sum_{i_1=1}^n \sum_{j_2=1}^n \sum_{i_2=1}^n c_{j_1 j_2} \cdot d_{i_1 i_2} \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} \quad (11)$$

La restricción reflejada en (9) se puede expresar mediante la función a minimizar (12). Y de forma análoga (10) se convierte en (13).

$$E_2 = \sum_{\substack{(i_1, j_1) \in \{1..n\}^2 \\ (i_2, j_2) \in \{1..n\}^2}} \delta(j_1, j_2) \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} - \sum_{(i, j) \in \{1..n\}^2} m_{ij} \quad (12)$$

$$E_3 = \sum_{\substack{(i_1, j_1) \in \{1..n\}^2 \\ (i_2, j_2) \in \{1..n\}^2}} \delta(i_1, i_2) \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} - \sum_{(i, j) \in \{1..n\}^2} m_{ij} \quad (13)$$

Siendo la función δ :

$$\delta(a, b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{en otro caso} \end{cases} \quad (14)$$

Para poder construir la red de Hopfield utilizando estas restricciones hay que realizar las operaciones necesarias para que las funciones de energía sean de la forma de una función de Lyapunov (8).

$$E_1 = -\frac{1}{2} \sum_{\substack{(i_1, j_1) \in \{1..n\}^2 \\ (i_2, j_2) \in \{1..n\}^2}} -c_{j_1 j_2} \cdot d_{i_1 i_2} \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} \quad (15)$$

$$E_2 = -\frac{1}{2} \sum_{\substack{(i_1, j_1) \in \{1..n\}^2 \\ (i_2, j_2) \in \{1..n\}^2}} -2 \cdot \delta(j_1, j_2) \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} + \sum_{(i, j) \in \{1..n\}^2} -m_{ij} \quad (16)$$

$$\begin{aligned}
E_3 = & -\frac{1}{2} \sum_{\substack{(i_1, j_1) \in \{1..n\}^2 \\ (i_2, j_2) \in \{1..n\}^2}} -2 \cdot \delta(i_1, i_2) \cdot m_{i_1 j_1} \cdot m_{i_2 j_2} \\
& + \sum_{(i, j) \in \{1..n\}^2} -m_{ij}
\end{aligned} \quad (17)$$

Una vez se han fijado las condiciones, la función de energía de la red se obtiene mediante una combinación lineal de las funciones de energía asociadas a cada restricción (18).

$$E = a \cdot E_1 + b \cdot E_2 + c \cdot E_3 \quad (18)$$

La elección de los parámetros a , b y c determina en qué medida se considera una condición más importante que otras. Para que las soluciones que se obtengan al aplicar la red sean válidas interesa que E_2 y E_3 tengan mucho más peso que E_1 . Una relación posible entre estos parámetros viene definida por la siguiente expresión.

$$\frac{b}{a} = \frac{c}{a} > 2 \cdot \max(d_{ij}) \quad (19)$$

Si se hace que a valga 1, entonces b es igual a c y los valores de los pesos (20) y el umbral (21) son:

$$\begin{aligned}
w_{(i_1, j_1)(i_2, j_2)} = & -c_{j_1 j_2} \cdot d_{i_1 i_2} \\
& -2 \cdot b \cdot (\delta(i_1, i_2) + \delta(j_1, j_2))
\end{aligned} \quad (20)$$

$$\theta_{(i, j)} = -2 \cdot b \quad (21)$$

Para aplicar este procedimiento a unos valores concretos de c y d basta con sustituirlos en las expresiones (20) y (21). Con el resultado obtenido para \mathbf{W} y θ se realiza el método de actualización secuencial, con una configuración inicial aleatoria para los valores de activación de las neuronas. El proceso se repite hasta que se alcance un estado estable. Si la solución obtenida no es aceptable se repite este último paso con otra configuración inicial.

5 RESULTADOS EXPERIMENTALES

Para la realización de las pruebas se ha definido un grafo de dependencias con 16 módulos a colocar en 16 huecos posibles. El esquema de la figura 3 representa dicho grafo.

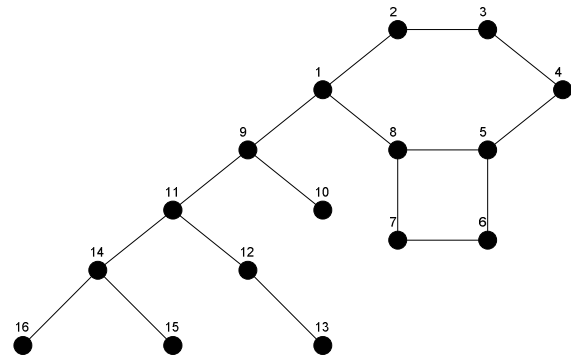


Figura 3: Grafo de dependencias.

Como se aprecia en la figura 3, la estructura a optimizar está formada por un bucle, que agrupa a los 8 primeros módulos, y por un árbol jerárquico. Esto permite observar el comportamiento del algoritmo para una configuración con distintas topologías. La matriz de dependencias correspondiente al grafo se muestra en (22).

$$C = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \quad (22)$$

Los huecos, en donde se sitúan los módulos, son todos del mismo tamaño y se disponen de forma uniforme dentro de un cuadrado. La numeración utilizada es la siguiente:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figura 4: Distribución de los huecos.

La distancia entre dos huecos se calcula sumando el número de filas y de columnas que les separan. Así, la matriz de distancias es la mostrada en (23).

$$D = \begin{pmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 & 3 & 2 & 3 & 4 & 4 & 3 & 4 & 5 \\ 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 & 4 & 3 & 2 & 3 & 5 & 4 & 3 & 4 \\ 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 5 & 4 & 3 & 2 & 6 & 5 & 4 & 3 \\ 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 & 3 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 & 4 & 3 & 2 & 3 \\ 4 & 3 & 2 & 1 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 5 & 4 & 3 & 2 \\ 2 & 3 & 4 & 5 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 & 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 3 & 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 \\ 5 & 4 & 3 & 2 & 4 & 3 & 2 & 1 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 \\ 3 & 4 & 5 & 6 & 2 & 3 & 4 & 5 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 \\ 4 & 3 & 4 & 5 & 3 & 2 & 3 & 4 & 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 \\ 5 & 4 & 3 & 4 & 4 & 3 & 2 & 3 & 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 \\ 6 & 5 & 4 & 3 & 5 & 4 & 3 & 2 & 4 & 3 & 2 & 1 & 3 & 2 & 1 & 0 \end{pmatrix} \quad (23)$$

Como la distancia máxima entre dos huecos es 6, se puede hacer que b valga 30. De esta forma los pesos y el umbral quedan de la siguiente forma:

$$w_{(i_1, j_1)(i_2, j_2)} = -c_{j_1 j_2} \cdot d_{i_1 i_2} - 60 \cdot (\delta(i_1, i_2) + \delta(j_1, j_2)) \quad (24)$$

$$\theta_{(i, j)} = -60 \quad (25)$$

Las siguientes figuras representan el estado de activación de las neuronas en las distintas pasadas del método de actualización secuencial de la red de Hopfield. El cuadro en negro indica un estado 1 y el blanco el 0. Las columnas representan a las distintas posiciones y las filas a los módulos.

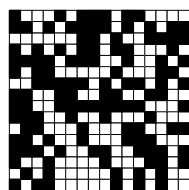


Figura 5: Activación inicial de las neuronas.

La figura 5 se corresponde con el estado inicial de las neuronas. El valor que se asigna a cada una de ellas se determina con un 50% de probabilidades de estar activa.

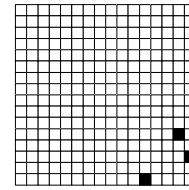


Figura 6: Activación después de una pasada.

En la figura 6 se puede observar que tras la primera iteración la mayoría de las neuronas han sido desactivadas. Esto se debe a la aplicación de las restricciones (9) y (10), que tienen una gran relevancia en la función de energía.

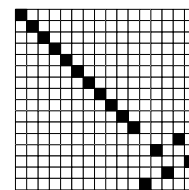


Figura 7: Estado estable tras dos pasadas.

Con sólo dos pasadas se ha conseguido alcanzar el estado estable que se muestra en la figura 7. En él se aprecia que los elementos relativos al bucle ocupan la diagonal principal, mientras que los pertenecientes al árbol se encuentran distribuidos en el resto de la matriz.

La simulación ha sido realizada en un equipo Pentium III a 700MHz con 128Mb de memoria RAM. El tiempo total invertido en la ejecución es inferior a 2 segundos.

Los valores de las energías que se obtienen en los distintos pasos son: (tabla 1)

Tabla 1: Energía de la red cada etapa.

Pasada	Energía de la red
Inicial	49258
1	-180
Estable	-929

Como se puede observar, en cada pasada se han realizado sólo los cambios de activación necesarios para que el valor resultante de la función de energía descendiera. Así pues, el estado estable se corresponde pues con un mínimo local de la función de energía. La convergencia de la red demuestra que esta función ha sido formulada de forma correcta.

La solución que se ha logrado en este caso se interpreta como se muestra en la figura 8. Las casillas son los huecos disponibles y el número mostrado se corresponde con el módulo que lo está ocupando.

1	2	3	4
5	6	7	8
9	10	11	16
13	15	12	14

Figura 8: Solución obtenida.

Estudiando minuciosamente la figura 8, se descubre que la parte correspondiente al bucle no ha sido optimizada de forma correcta. En experimentos sucesivos se comprobó que este comportamiento se mantiene. El que siempre se produzca una asignación de estos módulos a la diagonal principal se debe al orden preestablecido en que se recorre la matriz de neuronas en el proceso de activación selectiva, que en el ejemplo mostrado es siempre de izquierda a derecha por columnas y luego hacia abajo por filas.

6 CONCLUSIONES

Una vez se ha conseguido superar la dificultad de formular la función de energía a minimizar, la aplicación de la red de Hopfield al caso concreto es inmediata.

Por otro lado, para un número concreto de módulos, la complejidad del grafo de dependencias no afecta al rendimiento del algoritmo.

Como se puede apreciar en la simulación mostrada, en cada iteración se produce un cambio de estado en la neurona seleccionada sólo si con ello se reduce la función de energía.

Aunque se consigue una buena solución en un tiempo despreciable usando un equipo bastante limitado, el presente método presenta problemas a la hora de optimizar la distribución de módulos en bucle. El problema se debe a que la red cae en un mínimo local en el que un cambio de posición de un elemento del bucle no implica un descenso de la función de energía.

Así, en esta aplicación en concreto y con las funciones antes descritas, se obtiene una solución muy próxima a la óptima si los módulos se relacionan de forma jerárquica.

Una posible mejora al algoritmo propuesto consistiría en aplicar el enfriamiento simulado en la función de activación de las neuronas, convirtiendo la red de Hopfield en una máquina de Boltzmann. Con esto se permitirían cambios en el estado de las neuronas que no produjesen un descenso inmediato del valor de la función de energía al principio del proceso, permitiendo la salida de un posible mínimo local.

Otra forma de ampliar la búsqueda de la solución óptima es combinar la aproximación propuesta con una resolución mediante algoritmo genético, de forma que las distintas soluciones conseguidas con la red neuronal se utilicen para generar la mayor parte de la población [5].

Por último, el elevado consumo en memoria que se produce al construir la matriz de pesos puede llegar a ser un problema importante dependiendo de la cantidad de recursos disponibles en el equipo en que se ejecuta el algoritmo. Esta matriz crece de forma n^4 , siendo n el número de módulos a colocar.

En caso de estar limitados en espacio, se podrían calcular los valores de los pesos cada vez que se necesitasen si n fuera demasiado grande.

Otra posible solución a este problema sería aprovechar la simetría que existe dentro de la matriz obtener así una reducción en consumo de memoria del 50%.

Referencias

- [1] Hopfield, J. J., (1982) "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the USA*, vol. 79 no. 8 pp. 2554-2558.
- [2] Bharitkar. S. et al, (1999) "Microcode optimization with neural networks", *IEEE-NN*, vol. 10 no. 3 pp. 698.
- [3] Krusne, R., (2006) "Chapter 8: Hopfield networks", disponible en: <http://fuzzy.cs.uni-magdeburg.de/studium/nn/txt/folien060615.pdf>.
- [4] Xilinx Inc., (2002) "Virtex 2.5V Field Programmable Gate Arrays", disponible en: <http://direct.xilinx.com/bvdocs/publications/ds003-2.pdf>.
- [5] Xin Yao, (1992) "A Review of Evolutionary Artificial Neural Networks", Commonwealth Scientific and Industrial Research Organization, Australia.

CONTROL DEL PERÓXIDO DE HIDRÓGENO EN SISTEMAS SOLARES FOTO-FENTON

J.D. Álvarez

CIEMAT – Plataforma Solar de Almería, P.O. Box 22, 04200 Tabernas, Almería
 Universidad de Almería, Dpto. de Lenguajes y Computación, Área de Ingeniería de Sistemas y Automática,
 Crta. Sacramento s/n, 04120 La Cañada, Almería
 [josedomingo.alvarez@psa.es]

^{1,3}W. Gernjak, ¹S. Malato, ²M. Berenguel, ³M. Fuerhacker, ^{1,2}L.J. Yebra

¹ CIEMAT – Plataforma Solar de Almería, P.O. Box 22, 04200 Tabernas, Almería
 [wolfgang.gernjak@psa.es, sixto.malato@psa.es, luis.yebra@psa.es]

² Universidad de Almería, Dpto. de Lenguajes y Computación, Área de Ingeniería de Sistemas y Automática,
 Crta. Sacramento s/n, 04120 La Cañada, Almería [beren@ual.es]

³ University of Applied Life Sciences, Department of Water, Air and Environment, Muthgasse 18, 1190
 Vienna, Austria [maria.fuerhacker@boku.ac.at]

Resumen

El principal objetivo del proyecto CADOX es desarrollar una tecnología comercial híbrida de lazo cerrado basada en los Procesos de Oxidación Avanzada (POAs) para el tratamiento de aguas residuales industriales contaminadas por compuestos orgánicos persistentes. Entre ellos, el método Foto-Fenton es conocido por sus altos valores de reacción y su aplicabilidad con la luz natural del sol. El rango óptimo de la concentración del H_2O_2 durante el tratamiento Foto-Fenton depende del tipo y del grado de contaminación de las aguas residuales, una herramienta muy valiosa sería el disponer de un controlador automático que mantuviera la concentración deseada de H_2O_2 en cada reacción para reducir al mínimo la consumición del reactivo. Dicha concentración de H_2O_2 se ve afectada por la concentración de hierro en la disolución, la radiación incidente y la temperatura en el reactor. Para alcanzar los objetivos de control, se ha desarrollado un PI con anti-windup y un control anticipativo, a partir de un modelo lineal de la planta. Este trabajo resume los resultados obtenidos con estos controladores.

Palabras Clave: Procesos de Oxidación Avanzada, Tratamiento de aguas residuales, control de H_2O_2 , PID con anti-windup

1 INTRODUCCIÓN

El principal objetivo del proyecto CADOX [3], [6], [8] es desarrollar una tecnología comercial híbrida de lazo cerrado basada en los Procesos de Oxidación Avanzada (POAs) para el tratamiento de aguas

residuales industriales contaminadas por compuestos orgánicos persistentes [5], [10]. Entre ellos, el método Foto-Fenton es conocido por sus altos valores de reacción y su aplicabilidad con la luz natural del sol [4], [7].

Un esquema de la planta CADOX puede observarse en la figura (1).

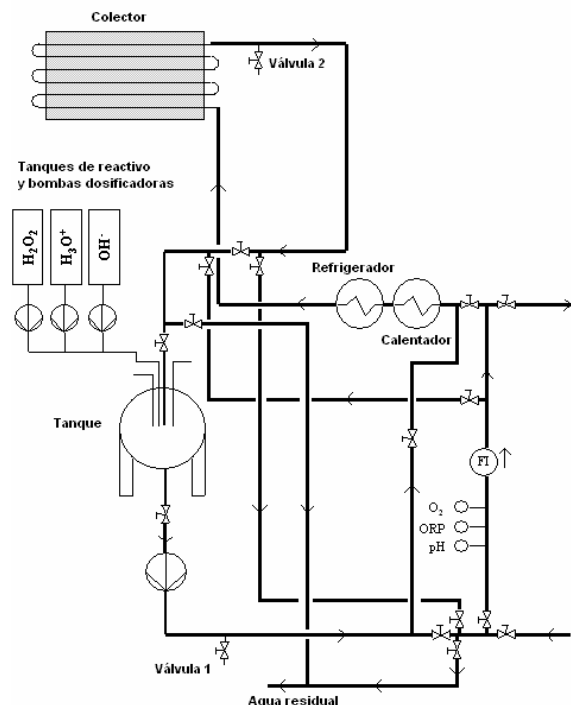


Figura 1: Esquema de la planta CADOX

Dentro de la planta CADOX se encuentran las siguientes herramientas utilizadas para su correcto funcionamiento:

- Un tanque de mezclado, donde se añadirá el H_2O_2 .
- Una bomba de recirculación, variando la velocidad de esta bomba se modificará el tiempo de recirculación de la disolución.
- Un colector, cuya superficie es de 4 m^2
- Resistencias térmicas para calentar la disolución.
- Un pequeño intercambiador de calor, donde un líquido refrigerante enfriará la disolución.
- Tres bombas dosificadoras.

Aparte, el sistema está complementado con un panel de instrumentación donde se encuentra toda la instalación eléctrica necesaria para el buen funcionamiento de sensores, bombas, válvulas... El PC de control enviará y recibirá señales de este panel.

El rango óptimo de la concentración de peróxido de hidrógeno (H_2O_2) durante el tratamiento Foto-Fenton depende del tipo y del grado de contaminación de las aguas residuales, en este sentido, un controlador automático que mantuviera la concentración deseada de H_2O_2 en cada reacción sería una herramienta muy valiosa para reducir al mínimo la consumición del reactivo. La concentración de H_2O_2 se ve afectada por la concentración de hierro en la disolución, la radiación incidente y la temperatura en el reactor. La concentración de H_2O_2 en disolución es controlada mediante una bomba con un regulador de frecuencia. El regulador de frecuencia permite ajustar la velocidad de la bomba para mantener la concentración de H_2O_2 cerca de la referencia deseada. Para alcanzar los objetivos de control un controlador PI con acción anti-windup y un control anticipativo, calculados a partir de un modelo lineal de la planta, fueron probados. El resultado de esos controladores probados en el proyecto CADOX es presentado en este trabajo.

2 MODELO DEL SISTEMA

Aunque el comportamiento del sistema es, en teoría, no lineal, se ha preferido hacer una aproximación lineal de éste, por los beneficios que aporta dicha aproximación en cuanto a simplicidad y rapidez. La ecuación global del sistema sacada en base a la respuesta de éste a una excitación de la entrada es mostrada en la ecuación (1)

$$y = c_{H_2O_2} = G_u(s) \cdot u + [G_I(s) \cdot I + G_P(s) \cdot c_{Fe}] \cdot P \cdot C \quad (1)$$

Cada uno de los términos de la ecuación pasan a definirse con más detalle a continuación:

- $c_{H_2O_2}$: salida del sistema, concentración de H_2O_2 en la disolución [mg/l]

- $G_u(s) = \frac{K}{s} \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} e^{-t_d s}$: función de transferencia que modela los efectos de un cambio de la frecuencia de la bomba dosificadora de H_2O_2 en la concentración de H_2O_2 .
- u : entrada del sistema (señal de control, frecuencia de la bomba dosificadora de H_2O_2 [%])
- $G_I(s) = -\frac{K_I}{s}$: función de transferencia que modela las pérdidas en la concentración de H_2O_2 debidas a la radiación solar incidente.
- I : radiación solar incidente en el colector [W]
- $G_P(s) = -\frac{K_P}{s}$: función de transferencia que modela las pérdidas en la concentración de H_2O_2 debidas a la concentración de hierro en dicha disolución.
- c_{Fe} : concentración de hierro en la disolución [mg/l]
- P : toma el valor uno en el caso de que haya hierro en la disolución, toma cero en el caso contrario [-].
- C : constante que depende del tipo de agua que se utilice durante el ensayo [-]

Como se ha comentado anteriormente la función de transferencia, $G_u(s)$, relaciona la concentración de H_2O_2 en la disolución con la frecuencia de la bomba dosificadora de H_2O_2 . Es fácil advertir que la bomba actúa como un integrador cuando añade H_2O_2 al tanque de mezclado, debido a que la bomba solo puede añadir H_2O_2 y no tiene manera de eliminarlo de la concentración, por lo que si se mantuviera un escalón constante en la bomba, la concentración de H_2O_2 en la disolución aumentaría, teóricamente, indefinidamente. Por otra parte, el comportamiento oscilatorio que se puede observar cuando se inyecta H_2O_2 a través de la bomba en el tanque de mezclado, es debido al bucle hidráulico del sistema, una vez que se inyecta H_2O_2 en el tanque de mezclado la mezcla no se vuelve instantáneamente homogénea, sino que la disolución tiene que recircular varias veces a través del sistema, el tiempo de recirculación es impuesto por un valor constante de flujo del fluido. Este comportamiento oscilatorio anteriormente comentado se encuentra bien modelado por un término de segundo orden. Por último un término de retraso en la salida del sistema debido a la posición del sensor de medida en la planta es añadido para completar la ecuación. Este tipo de comportamiento descrito también ha podido ser observado mientras se intentaba controlar el nivel de pH en fotobioreactores [2]. En la siguiente figura se muestra una gráfica comparando la salida del modelo previamente descrito contra la salida del sistema:

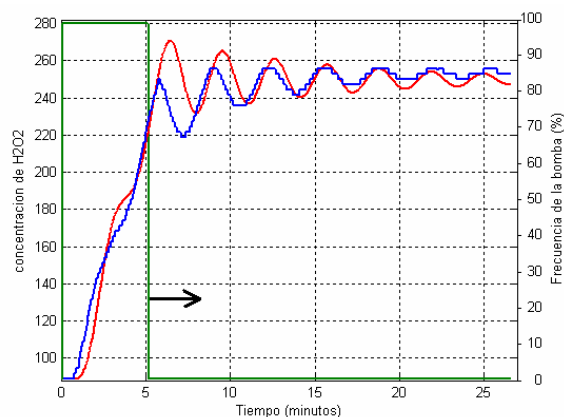


Figura 2: Salida del sistema comparada con una simulación del modelo

En la gráfica se pueden observar tres señales:

1. Referencia de la concentración de H_2O_2 (verde).
2. Nivel de concentración de H_2O_2 en la planta (azul)
3. Salida simulada por el modelo (rojo).

El sistema se excitó con un impulso, debido a su comportamiento integrador, como se observa en la figura después del retraso inicial, la señal del modelo consigue ajustarse con precisión a la salida real de la planta. Las oscilaciones que se observan tanto en la señal simulada, como en la salida de la planta, corresponden a la recirculación que sufre la disolución en la planta hasta que la mezcla se vuelve homogénea.

Destacar también que las funciones de transferencia $G_p(s)$ y $G_I(s)$ modelan, respectivamente, el consumo de H_2O_2 en la disolución debido a la concentración de hierro existente en la disolución y a la radiación solar incidente en el colector, aunque estas relaciones también son descritas mediante ecuaciones no-lineales también han sido simplificadas en modelos lineales. La función de transferencia $G_p(s)$ que relaciona el consumo de H_2O_2 con los mg/l de hierro existentes en la disolución es definida como un integrador con una ganancia estática negativa, ya que cuanto mayor sea la concentración de hierro mayores serán las pérdidas de H_2O_2 en la disolución. Con el mismo tipo de función de transferencia se puede relacionar el consumo de H_2O_2 en la disolución con los vatios incidentes en la superficie del colector.

En la figura (3) se puede ver una comparación de la salida de los modelos de consumo de H_2O_2 contra un ensayo realizado en la planta, la señal verde correspondería a la salida real del sistema durante el ensayo, mientras que la señal azul corresponde a la salida de los modelos de pérdidas de H_2O_2 . Para obtener dicha señal es necesario sumar la salida de las funciones de transferencia $G_p(s)$ y $G_I(s)$. En el

ensayo había una concentración de 1.35 mg/l de hierro en la disolución, al principio todo el colector estaba cubierto y aun así había un poco de consumo de H_2O_2 , a partir del minuto 155 aproximadamente, se destapa totalmente el colector y en la gráfica se puede observar como el consumo de H_2O_2 aumenta rápidamente. Para finalizar el experimento 20 minutos después se vuelve a cubrir el colector lo que provoca que el consumo de H_2O_2 se suavice. Como se observa en la gráfica el modelo es capaz de reproducir fielmente el consumo de H_2O_2 en el sistema debido a la concentración de hierro o a la radiación incidente.

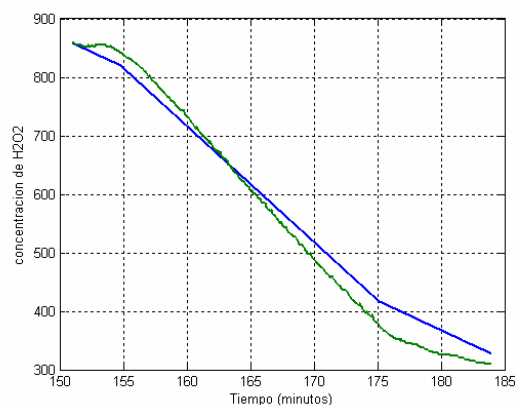


Figura 3: Ensayo para comprobar los modelos de consumo de H_2O_2

Finalmente hay que tener en cuenta que las consideraciones hechas anteriormente para factores tales como la concentración de hierro en la disolución o la radiación incidente son también válidos para la temperatura de la disolución, ésta también tiene influencia en el consumo de H_2O_2 y se podría modelar por una función de transferencia parecida a las anteriores, llamada por ejemplo $G_T(s)$, aunque este factor no ha sido tenido en cuenta en este trabajo debido a que la temperatura se mantuvo constante durante los ensayos, por ese motivo no aparece ninguna función de transferencia que relacione la temperatura de la disolución con el consumo de H_2O_2 en la ecuación (1).

3 DISEÑO DE UN CONTROLADOR PARA EL SISTEMA

3.1 CONSIDERACIONES PREVIAS

El sistema de control más sencillo que se ha pensado es capaz de controlar con solvencia el sistema descrito en la anterior sección es un controlador PI con una acción *anti-windup* [1] para evitar que la señal que el controlador le envía a la bomba dosificadora exceda del rango de sus capacidades físicas (0% - 100%) debido a la acción integral del

propio controlador. La acción *anti-windup* básicamente cambia el término integral del controlador para evitar que el sistema entre en saturación, previniendo de esta manera diferencias entre la salida del controlador y la salida real que el actuador (en este caso la bomba dosificadora de H_2O_2) le envía al sistema (en este caso la planta CADOX).

Además del controlador PI que se ha comentado en el párrafo anterior, un controlador anticipativo (*feedforward*) ha sido añadido con el objetivo de mejorar el rendimiento total del sistema de control. El control anticipativo fue desarrollado usando las ganancias estáticas de los modelos lineales de consumo de H_2O_2 $G_p(s)$ y $G_I(s)$ descritos en el párrafo anterior. Ya que la temperatura de la disolución se mantuvo constante durante la operación de la planta, no fue necesario el desarrollo de un controlador anticipativo equivalente para compensar el consumo de H_2O_2 debido a cambios en la temperatura, $G_I(s)$.

3.2 PROBANDO EL SISTEMA DE CONTROL

Una vez que el sistema de control ha sido diseñado, fue probado durante un ensayo en la planta CADOX con el objetivo de verificar todo el desarrollo teórico del modelo comentado en la anterior sección. Este ensayo puede observarse en la figura (4).

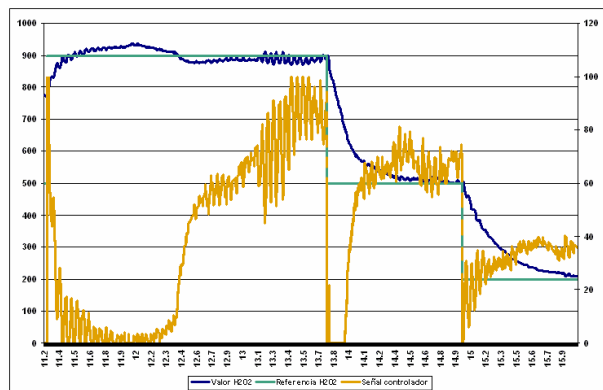


Figura 4: Ensayo para probar el control

En la figura (4) se pueden observar tres señales:

1. La referencia o consigna de H_2O_2 .
2. El valor actual de la concentración de H_2O_2 .
3. La señal que envía el actuador (la bomba dosificadora) al sistema (la planta CADOX).

Durante el ensayo la concentración de hierro era de 1 mg/l, concentración que se le añadió bruscamente una vez empezado el ensayo para ver lo bien que respondía el controlador, y que iría disminuyendo a lo largo del ensayo al consumirse junto al H_2O_2 .

Después de añadirle hierro a la concentración el colector se descubrió, lo que provoca otro cambio brusco en el consumo de H_2O_2 , el resto del tiempo el colector se mantuvo descubierto con lo que las únicas variaciones existentes en la radiación solar incidente podrían ser pasos de nubes, que no hubo ninguno durante la operación, o el cambio normal que sufre la radiación durante el paso del día. Por último la temperatura de la disolución durante el ensayo fue mantenida constante.

La referencia inicial de concentración de H_2O_2 fue establecida en 900 mg/l, un valor muy alto para una operación normal, pero que se probó debido a que se quería ver lo bien que el modelo funcionaba en condiciones 'extremas', se puede ver como el valor de H_2O_2 excede por poco a la referencia deseada, debido a dos factores: que el modelo no fue diseñado para ese rango de operación y a que al no estar el colector descubierto, ni haber hierro en la disolución no hay manera de disminuir la concentración de H_2O_2 .

Poco después y como se ha comentado anteriormente, se le añadió a la disolución 1 mg/l de hierro con el colector cubierto para que hubiera un aumento del consumo de H_2O_2 y observar así lo bien que el controlador se adaptaba a dichos cambios. Como se puede observar en la gráfica el consumo empieza a aumentar, lo que provoca que la concentración disminuya.

Antes de que alcance la referencia deseada de 900 mg/l se destapa completamente el colector, hecho que provoca que el consumo aumente bruscamente, con lo que el valor de H_2O_2 cae un poco por debajo de la referencia. Aun así el controlador es capaz de mantenerlo cerca de la referencia y acercarlo a ella progresivamente. Sin embargo cuando está cerca de alcanzar a la referencia se producen perturbaciones por causas desconocidas, aunque el controlador sigue siendo capaz de ir llevando el valor de H_2O_2 a la referencia deseada.

Una vez que el valor de H_2O_2 está tan cerca de la referencia como se desea y se ha estabilizado, se cambia el valor de la referencia a 500 mg/l. Este valor es un valor de operación más normal que el setpoint anterior, por lo que el controlador es capaz de alcanzarlo con más facilidad y sin tantas oscilaciones.

Por último, una vez que el valor de H_2O_2 se hubo estabilizado cerca de la referencia deseada de 500 mg/l, se volvió a cambiar dicha referencia a un nuevo valor de 200 mg/l. Como en el caso anterior el controlador es capaz de llevar el sistema hasta la referencia deseada y todavía con menos oscilaciones que en el caso anterior debido a que esta referencia

estaba más próxima al punto de operación para el que se calculó el modelo lineal de la planta.

Como se puede observar en la gráfica, aunque los resultados son aceptables, no son tan buenos como los teóricos esperados debido a errores de modelado. Estos errores de modelado son debidos tanto a la aproximación lineal del sistema, como a la aproximación lineal de los modelos de pérdidas. Además hay ligeras fluctuaciones en la concentración de H_2O_2 cuando ésta se encuentra cerca de la referencia, aunque nunca exceden de 20 mg/l.

4 CONCLUSIONES

Este trabajo trata sobre el desarrollo de un sistema de control para la planta CADOX, basado en un modelo lineal simplificado de dicha planta. El sistema de control está formado por un controlador PI con acción *anti-windup* y un controlador anticipativo.

El controlador anticipativo fue añadido para mejorar el control de la concentración de H_2O_2 anticipándose a cambios en el consumo de dicha concentración debidos a la concentración de hierro o a la radiación incidente en el colector. Aunque el control anticipativo causa una respuesta más rápida del sistema de control también implica un comportamiento oscilatorio en el valor de concentración de H_2O_2 .

Este comportamiento oscilatorio es probablemente debido a una aproximación matemática de los modelos de pérdidas, por concentración de hierro y por radiación solar incidente, no del todo satisfactoria, ya que aunque dicha influencia es no lineal, un modelo lineal aproximado fue utilizado por los beneficios que este tipo de modelos otorga en cuanto a rapidez y simplicidad, usando para su cálculo sólo condiciones estacionarias en el término anticipativo.

Agradecimientos

Los autores de este trabajo desean dar las gracias a la Comisión Europea por el soporte financiero para el proyecto CADOX (Contrato no. EVK1-CT-2002-00122), a los fondos CICYT-FEDER (proyectos DPI2002-04375-C03 y DPI2004-07444-C04-04) y a la Junta de Andalucía Consejería de Innovación, Ciencia y Empresa. Este trabajo también se ha realizado en el ámbito del acuerdo específico de colaboración entre la Plataforma Solar de Almería y el Grupo de Investigación "Automática, Electrónica y Robótica" de la Universidad de Almería (TEP197) titulado "*Desarrollo de sistemas y herramientas de control para plantas termosolares*".

Referencias

- [1] Åström, K.J., y Wittenmark, B., (1984) *Computer Controlled Systems*, Prentice-Hall.
- [2] Berenguel, M., Rodríguez, F., Acién, F.D, y García, J.L., (2004) "Model predictive control of pH in tubular photobioreactors", *J. Proc. Control*, 14, pp. 377-387.
- [3] Blanco, J., Malato, S., Fernández, P., Vidal, A., Morales, A., Trincado, P., Oliveira, J.C., Minero, C., Musci, M., Casalle, C., Brunotte, M., Tratzky, S., Dischinger, N., Funken, K.H., Sattler, C., Vincent, M., Collares-Pereira, M., Mendes, J. F., y Rangel C.M., (2000) "Compound parabolic concentrator technology development to commercial solar detoxification applications", *Sol. Energy*, 67, pp. 317-330.
- [4] Fallmann, H., Krutzler, T., Bauer, R., Malato, S., y Blanco, J., (1999) "Applicability of the photo-Fenton method for treating water containing pesticides", *Catal. Today*, 54, pp. 309-319.
- [5] Legrini, O., Oliveros, E., y Braun, A.M., (1993) "Photochemical Processes for Water Treatment", *Chem. Rev.*, 93, pp. 671-698.
- [6] Malato, S., Blanco, J., Vidal, A., y Richter, C., (2002) "Photocatalysis with solar energy at a pilot-plant scale: an overview", *Appl. Catal. B: Environ.*, 37, pp. 1-15.
- [7] Malato, S., Blanco, J., Vidal, A., Alarcón, D., Maldonado, M.I., Cáceres, J., y Gernjak, W., (2003) "Applied studies in solar photocatalytic detoxification: an overview", *Sol. Energy*, 75, pp. 329-336.
- [8] Malato Rodríguez, S., Blanco Gálvez, J., Maldonado Rubio, M.I., Fernández-Ibañez, P., Alarcón Padilla, D., Collares Pereira, M., Farinha Mendes, J., y Correia de Oliveira J., (2004) "Engineering of solar photocatalytic reactors", *Sol. Energy*, 77, pp. 513-524.
- [9] Ogunnaike B.A. y Ray W. H., (1994) *Process Dynamics, Modeling, and Control*, Oxford University Press.
- [10] Safarzadeh-Amiri, A., Bolton, J.R., y Cater, S.R., (1996) "The use of iron in advanced oxidation processes", *J. Adv. Oxid. Technol.*, 1, pp. 18-26.

CONTROL JERÁRQUICO BORROSO PARA UN CAMPO DE COLECTORES SOLARES DISTRIBUIDOS

C.M. Cirre^{1,2}, L. Valenzuela¹, M. Berenguel²

¹Plataforma Solar de Almería, Ctra. de Senés s/n, E-04200, Almería, España

²Universidad de Almería, Dpto. Lenguajes y Computación, Ctra. Sacramento s/n, 04210, Almería, España
cristina.martinez@psa.es

Resumen

Este artículo muestra un esquema de control jerárquico para regular la temperatura de salida del fluido de trabajo en un campo de colectores solares distribuidos y calcular la temperatura de referencia de dicho fluido. Dicha planta se encuentra expuesta a fuertes perturbaciones y no linealidades que dificultan su operación de forma manual, tanto para la regulación como para escoger el punto de operación adecuado. Se ha implementado un algoritmo de cálculo automático de consignas utilizando la lógica borrosa con la finalidad de encontrar la temperatura de referencia adecuada que maximice la temperatura de salida y cumpla las restricciones de operación de la planta.

Palabras clave: control jerárquico, control por adelanto, control IPD, energía solar

1. INTRODUCCION

El campo de colectores solares distribuidos ACUREX, en el cual se ha realizado este trabajo, ha sido utilizada durante varios años como planta piloto para ensayar distintos tipos de algoritmos de control y más tarde ser aplicados en futuras instalaciones de sistemas de concentración solar cilindro-parabólica. Se han experimentado las técnicas de control más innovadoras a lo largo de los años, como las basadas en control predictivo [5] y también en lógica borrosa [3],[4],[6],[9],[15]. Hasta ahora, los esfuerzos por mejorar los algoritmos de control se centraban en obtener una respuesta lo más rápida posible con la menor oscilación ante un cambio de consigna, potenciando así la velocidad de respuesta del sistema junto con la eficacia del rechazo de perturbaciones. Sin embargo, en los procesos industriales en general, la tendencia es la de controlar cada vez sistemas más complejos que necesitan ser jerarquizados, mediante un sistema de jerarquías, con un sistema supervisor en cada nivel [1]. Objetivos a tener en cuenta durante la operación de la planta como los de mantener las restricciones de seguridad de la planta o iniciar el arranque de la operación del campo de colectores solares eran aún desarro-

llados de forma manual por el operador de esta planta.

Previamente a este trabajo, se desarrolló un generador de consignas basado en modelo físico para la misma planta, obteniéndose buenos resultados [7][11], sobre todo en la fase de arranque de operación de la planta. Sin embargo, a pesar de que tanto la radiación solar tenida en cuenta por el generador como la salida calculada por el mismo eran filtradas, se observó que el generador de consignas basado en modelo físico era muy sensible a las variaciones de radiación solar, de forma que, cuando las variaciones de la radiación solar por el efecto de las nubes eran significativas y continuadas, provocaba grandes variaciones en la temperatura de referencia calculada no deseadas. Con el objetivo de desarrollar una estructura jerárquica que no utilice el modelo físico de la planta, el cual es muy sensible a las variaciones de la radiación solar, y teniendo en cuenta la similitud entre el razonamiento humano y la lógica borrosa, se ha desarrollado un generador de consignas basado en lógica borrosa (GCMB) para el campo de colectores solares distribuidos. El GCMB ocupa la segunda capa de una estructura jerárquica de control, donde la capa de regulación (capa 1) estaría formada por un control por adelanto y un control IPD (Control Integral-Proporcional-Derivativo).

2. LA PLANTA DE COLECTORES SOLARES DISTRIBUIDOS

El campo de colectores solares distribuidos ACUREX se encuentra situado en la Plataforma Solar de Almería, en el desierto de Tabernas (Almería). El campo solar consta de 480 módulos de colectores solares cilindro-parabólicos que se encuentran distribuidos en 10 filas paralelas (figura 1). La finalidad de cada colector es la de concentrar la radiación solar directa (I_b , $W \cdot m^{-2}$) en el receptor situado en la línea focal de la parábola, por el cual, internamente, se encuentra circulando un fluido de transferencia térmica (figura 2). Dicho fluido es un tipo de aceite sintético (Santotherm

55) que puede ser calentado hasta 300 °C sin peligro de degradarse.



Figura 1: Campo de colectores solares distribuidos ACUREX y tanque de almacenamiento térmico

El proceso consiste en extraer el aceite de la parte baja del tanque (donde reside el aceite a más baja temperatura) para ser impulsado por la bomba de impulsión (figura 2) hacia el campo solar, de forma que a su paso por el mismo, el aceite es calentado, en mayor o menor medida dependiendo principalmente de cuatro factores: la velocidad a la que circula el fluido ($q, l \cdot s^{-1}$), la radiación solar corregida ($I, W \cdot m^{-2}$) calculada en función del ángulo de incidencia de la radiación solar directa, I_b , con respecto a los espejos, la temperatura del aceite que proviene del tanque ($T_{in}, ^\circ C$) y la temperatura ambiente ($T_{amb}, ^\circ C$). La señal de control que se puede manipular para regular la temperatura de salida ($T_{out}, ^\circ C$) es la velocidad del fluido q . El resto de variables no pueden ser manipuladas y por lo tanto son consideradas como perturbaciones, que dificultan considerablemente la operación de la planta. Otros elementos que dificultan su operación es la dinámica no lineal de la misma (con ganancias, retardos y constantes de tiempo variables) así como las restricciones de operación que deben cumplirse para mantener en buen estado los elementos que la forman. Dichas restricciones comprenden el no superar los límites físicos de la bomba (entre 2 y 10 $l \cdot s^{-1}$), siendo los caudales altos los que suponen un mayor esfuerzo de empuje para la bomba y no superar incrementos de temperatura mayores de 70 u 80 °C ($\Delta T = T_{out} - T_{in}$) entre la temperatura del aceite que entra y sale del campo, con la finalidad de evitar estrés en los materiales.

Por ello, es importante tener un sistema de control automático que controle la temperatura de salida y que regule q para alcanzar la temperatura deseada ($T_{ref}, ^\circ C$) de forma segura con las menores oscilaciones posibles. Una parte importante de la operación de la planta consiste en determinar un valor de T_{ref} . Hasta ahora, era determinada cada cierto tiempo, por el operador de la planta, basándose en la experiencia de operación de la misma. Sin embargo, situaciones que requieren cambios rápidos de T_{ref} para preservar las condiciones de seguridad de la planta (como la fase de arranque) o donde la cantidad de radiación solar es insuficiente para alcanzar los 70 °C máximos dificultan la elección de una T_{ref} adecuada por parte del operador. Desde el punto de vista de control, se puede controlar cualquiera de las temperaturas de salida de los lazos de colectores, la máxima de todas o la media. En nuestro caso, se ha escogido controlar el lazo número 2.

La finalidad de este campo de colectores solares es la de calentar la mayor cantidad de aceite y almacenarlo para después ser utilizado en procesos de generación de electricidad o en procesos de desalinización.

3. CONTROL JERÁRQUICO BASADO EN MODELO BORROSO

Como se ha comentado en la sección anterior, las distintas situaciones de operación de la planta dificultan en ocasiones el trabajo del operador, obteniéndose un menor rendimiento del proceso y favoreciendo posibles daños materiales en los ele-

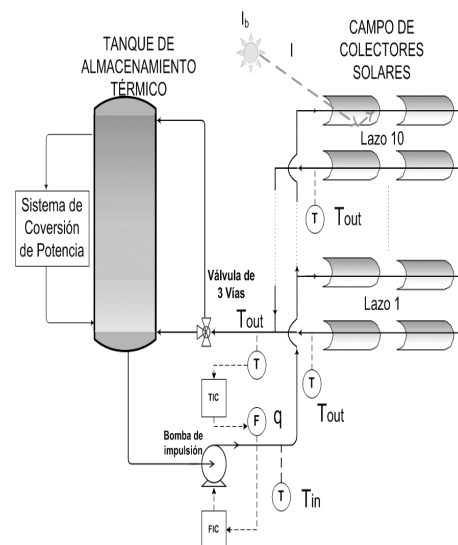


Figura 2: Esquema de señales y medidas más significativas del campo solar

mentos que componen la planta. En este tipo de plantas, la implementación de un sistema de regulación de la temperatura de salida se justifica por la necesidad de precisión en dicha temperatura de salida (a pesar de las perturbaciones que afectan a la planta) y alcanzar dichas temperaturas en un período de tiempo lo suficientemente corto para evitar situaciones de estrés indeseados en los materiales. La mejora del rendimiento de la planta a través de un generador de consignas conlleva cumplir los siguientes objetivos: establecer un valor de T_{ref} alcanzable de acuerdo a las condiciones actuales de la planta (radiación solar, temperatura de entrada, ...etc.); que dicha consigna cumpla las restricciones de seguridad de la planta; que ayude a los algoritmos de control en períodos de operación como el arranque y parada del sistema; evitar puntos de operación críticos que pongan en peligro la actuación de los controladores (por ejemplo bajos caudales cercanos al límite de saturación) y llevar la operación de la planta a un punto de operación de forma segura.

Para cumplir con todos estos objetivos, se ha desarrollado un control jerárquico multicapa formado por un generador de consignas basado en lógica borrosa, un control por adelantado y un control tipo IPD (figura 3).

3.1. CONTROL POR ADELANTO Y CONTROL IPD

La primera capa de la estructura de control jerárquica se corresponde con la capa de regulación. Está formada por un controlador por adelantado (1) y un control IPD en paralelo. Los algoritmos de control por adelantado son ampliamente utilizados en la industria, sobre todo en plantas sometidas a numerosas perturbaciones como en el ACUREX [5]. Su objetivo es la de calcular la señal manipulada (en nuestro caso q_{ff} , $l \cdot s^{-1}$, en figura 3) en función de las perturbaciones medibles que afectan a la planta, anticipándose al cálculo de la señal de control. Este tipo algoritmos no se encuentran realimentados, por lo que, en muchas ocasiones, debido a los errores de modelado o de ajuste de parámetros (como en este caso el de la reflectividad de los espejos) no suelen corregir el error en estado estacionario. Es habitual incorporar a este algoritmo otra estructura de control que se encuentre realimentada y pueda corregir este tipo de error. En este caso se ha escogido utilizar una estructura de control tipo IPD previamente utilizada en otros trabajos en los cuales ha demostrado un buen comportamiento [7]. Con el caudal calculado por el IPD (q_{ipd} , en $l \cdot s^{-1}$) junto con el calculado por el control por adelantado se obtiene el caudal necesario para realizar el seguimiento de la T_{ref} de forma adecuada.

$$q_{ff} = \frac{\alpha \cdot I \cdot G \cdot L - L \cdot (1.12 \cdot (T_{med} - T_{amb}) - 19.8)}{\rho(T_{med}) \cdot \Delta h} \quad (1)$$

$$\Delta h = c_p(T_{med}) \cdot (T_{out} - T_{in}) \quad (2)$$

$$d_{tin} = A_1 \cdot e^{-\left(\frac{q}{t_1}\right)} + y_0 \quad (3)$$

$$\rho(T_{med}) = 903 - 0.672 \cdot T_{med} \quad (4)$$

$$c_p(T_{med}) = 1820 - 3.478 \cdot T_{med} \quad (5)$$

En (1) se muestra la función de control utilizada para calcular el caudal necesario para alcanzar la temperatura de salida determinada por el GCMB (control por adelantado). Esta función se ha obtenido de la simplificación de las ecuaciones de balance de energía [5]. La salida del control por adelantado es el caudal q_{ff} (en esta ecuación en $m^3 \cdot s^{-1}$) calculada para el lazo 2. G es la apertura del colector (1.83 m). El término α es el producto de la eficiencia óptica del colector por la reflectividad actual de los espejos. Este valor debe ser medido externamente y actualizado cada cierto tiempo en el sistema de adquisición de datos. L es el valor de la longitud de un lazo (172 m). T_{med} es la temperatura media calculada entre la temperatura de salida actual y la temperatura de entrada. Las funciones $\rho(T_{med})$ y $c_p(T_{med})$ se corresponden con la función de densidad del aceite ($kg \cdot m^{-3}$) (ecuación 4) y el calor específico ($J \cdot kg^{-1} \cdot ^\circ C^{-1}$) (ecuación 5) del fluido de transferencia térmica. En (2) se muestra el incremento de entalpía, Δh , entre la temperatura de entrada al campo y la temperatura de salida. Se ha implementado una función empírica (obtenida por métodos experimentales con un coeficiente de determinación de 0.99) (3) para incluir en el esquema del control por adelantado el retardo existente debido al tiempo de residencia del fluido desde que es medido a la entrada del campo (T_{in}) hasta que llega a la salida del campo (T_{out}). Esta función depende de la velocidad del fluido y del lazo controlado. Los coeficientes A_1 , t_1 y y_0 varían según el lazo que es controlado. Los coeficientes para el lazo 2 son los siguientes: $A_1 = 2008.72$ (s), $t_1 = 1.82$ ($l \cdot s^{-1}$) e $y_0 = 184.36$ (s). Para sintonizar el IPD se han utilizado las reglas del mínimo ITAE [13]. Los parámetros del IPD fueron sintonizados con los siguientes valores: $K_p = -0.2$ ($^\circ C \cdot s \cdot l^{-1}$), $T_i = 200$ (s), $T_d = 70$ (s), diseñados para la zona de operación nominal de la planta.

3.2. GENERADOR DE CONSIGNAS BORROSO

En base a la experiencia de operación del campo de colectores solares, se puede decir que los dos parámetros que ejercen más influencia en la determinación del valor de T_{ref} adecuado son la radiación solar y la temperatura de entrada. La primera, nos va a determinar cuál es el incremento máximo de temperatura que se puede obtener en el campo solar, y la segunda nos indicara la temperatura que tiene el aceite justo a la entrada del campo solar, y por lo tanto, cual es la máxima que puede alcanzar sin exceder las restricciones de seguridad. Otra posibilidad hubiera sido sustituir el conjunto borroso de entrada T_{in} al GCMB por el nivel de caudal, que realmente es la variable que más influye en la no linealidad de la planta [10]. Sin embargo, resulta más fácil e intuitivo utilizar como segunda entrada T_{in} para poder diseñar el mecanismo de seguridad ΔT . También sería aceptable incluir como tercera entrada q , pero en una primera aproximación, se pretende conseguir un modelo simplificado.

En el caso de la radiación solar, el universo de discurso abarca todo el rango significativo de niveles de radiación, siendo $0 \text{ W} \cdot \text{m}^{-2}$ la cantidad mínima fijada y $1100 \text{ W} \cdot \text{m}^{-2}$ la cantidad máxima. La función de pertenencia tiene forma triangular, siendo el extremo izquierdo de tipo L y en el extremo derecho de tipo Γ (figura 4). En una primera versión, el número de conjuntos en los que se dividió el universo de discurso de la radiación solar fue más elevado, pero el comportamiento cuando había nubes era peor, haciendo un GCMB más sensible a estas variaciones, y por lo tanto, ejerciendo un efecto indeseado en el cálculo de T_{ref} . Reduciendo el número de conjuntos a tres niveles (con los siguientes valores lingüísticos: Bajo: B, Medio: M y Alto: A) los resultados en simulación [2] mejoraron.

Para el universo de discurso de la temperatura de entrada, el tipo de función de pertenencia escogido es también triangular. Los conjuntos se han definido de forma que se cumplan las restricciones de seguridad de la planta, en intervalos de 33°C en la mayoría. La división en conjuntos se puede ver en la figura 4. Los valores lingüísticos son: Muy Bajo (MB), Bajo (B), Bajo-Medio (BM), Medio (M), Alto (A), Muy Alto (MA). En el caso de la variable lingüística de salida T_{ref} , la función de pertenencia de tipo L es mayor ya que es la zona de temperaturas que están relacionadas con el arranque, y es una fase de dinámica rápida en la que no es necesaria la precisión. El resto de conjuntos borrosos son el resultado del mapeado entre los conjuntos de la temperatura de entrada y los

80°C de ΔT necesarios para mantener las restricciones de seguridad de la planta en incrementos de temperatura. Estos conjuntos borrosos se pueden observar en la figura 4 donde los valores lingüísticos con los de T_{in} .

La figura 5 muestra la superficie generada por el generador de consignas borroso, niveles de radiación desde 0 hasta 1100 en incrementos de $27.5 \text{ W} \cdot \text{m}^{-2}$ y desde 15°C hasta 250°C en intervalos de 6.7°C . Los incrementos en la temperatura de referencia eran aproximadamente de 3.65°C . Como se puede observar en la gráfica, la T_{ref} se incrementa conforme se incrementa la I y la T_{in} . Esto es lógico, si hay más cantidad de energía solar, es posible alcanzar mayor temperatura a la salida del campo. Las restricciones de la planta obligan a imponer un máximo de 80°C , pero esto es tenido en cuenta por el generador de consignas borroso al ser diseñado con un mapeado de los conjuntos de temperatura de entrada y al establecerlo en la base de reglas. La tabla 1 muestra la base de conocimiento diseñada en la que se basarán las reglas borrosas que utilizan el operador *max-min* [12][14] para ser calculadas. La base de conocimiento ha sido diseñada basándose en la experiencia de operación de la planta. Las relaciones fueron modificadas sobre todo, en la fase de arranque, para que las restricciones de la planta se cumplieran.

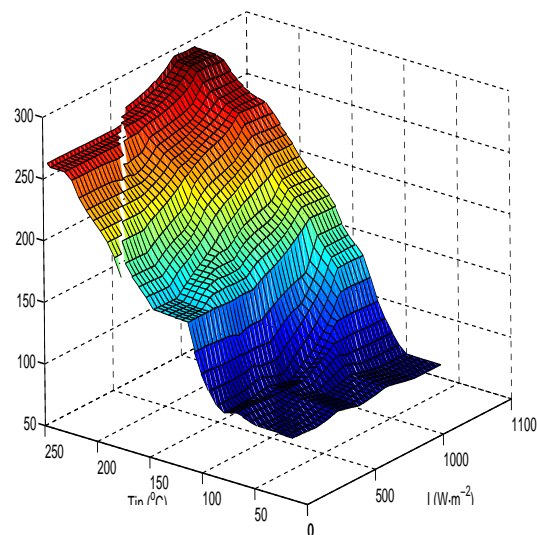


Figura 5: Superficie de T_{ref} generada por los universos del discurso de I y T_{in} para diferentes intervalos

Si se observa la figura 5 para un punto de la superficie en el que T_{in} es baja e I es muy alta, T_{ref} no alcanza valores demasiado altos, ya que si fuera muy alta no cumpliría las normas de seguridad.

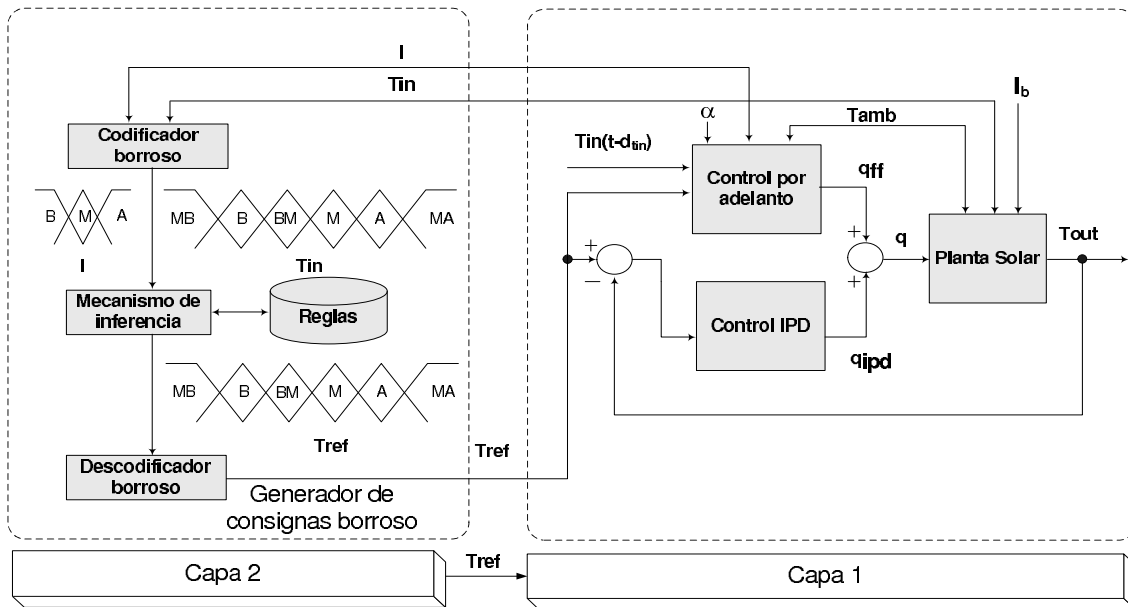


Figura 3: Esquema de control jerárquico multicapa formado por un generador de consignas borroso, un control por adelanto y un control IPD

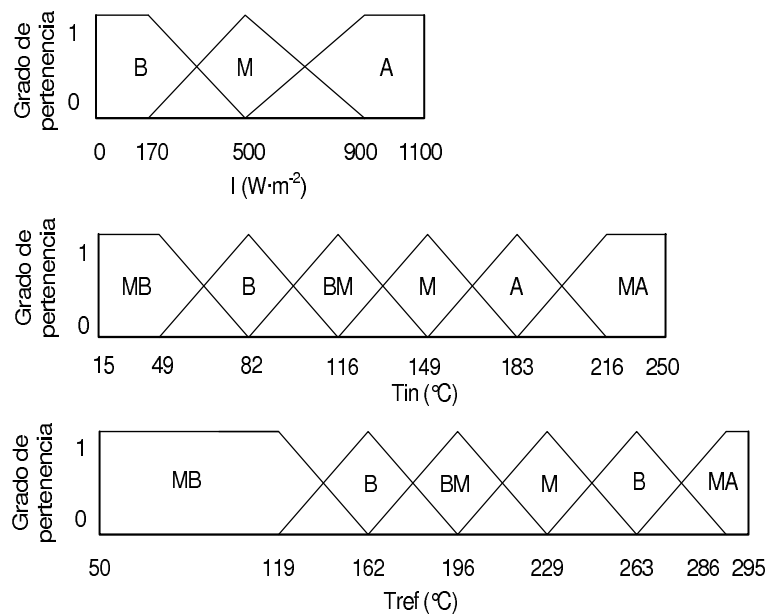


Figura 4: Conjuntos borrosos de entrada y salida para el GCMB

Tabla 1: Base de conocimiento que relaciona los valores lingüísticos de la temperatura de entrada y radiación solar corregida con las temperaturas de referencia.

I	T_{in}					
	VL	L	LM	M	H	VH
L	VL	VL	L	L	LM	H
M	VL	VL	L	LM	M	H
H	VL	L	LM	M	H	VH

4. RESULTADOS EXPERIMENTALES

En esta sección se van a mostrar varios experimentos desarrollados con la estructura jerárquica completa mostrada en la figura 3. El algoritmo ha sido implementado en código *C* [8] y adaptado al sistema de adquisición de datos del campo solar. El tiempo de muestreo escogido ha sido de 39 (s) y lazo de colectores controlado es el número 2. Antes de ser probada en la planta real, la estructura de control fue probada en simulación, obteniéndose resultados aceptables.

El objetivo que se pretende conseguir con los ensayos es demostrar que el GCMB calcula en cada instante la T_{ref} adecuada según las condiciones actuales y se mantiene dentro de las normas de seguridad.

La figura 6 muestra un ensayo de un día completo de operación. La parte que merece ser comentada en este ensayo es la fase de arranque. Se puede comprobar como evoluciona T_{ref} en función del incremento que se produce en T_{in} , demostrándose así que se cumplen las normas de seguridad de la planta. La operación normal del sistema hubiera requerido la atención completa del operador para estimar el tiempo de subida de la T_{in} y así, ir cambiando cada pequeños periodos de tiempo el valor de T_{ref} . Sobre las 12.75 horas en la misma figura, se produce un incremento muy rápido de T_{in} (de 60 °C aproximadamente en menos de media hora) lo cual hubiera representado un escalón de T_{ref} muy elevado en la operación manual, pero mediante el GCMB este incremento se produce de forma suave y continuado.

La figura 7 muestra un ensayo realizado en un día nublado. La función interesante del GCMB es la de evitar cambios bruscos en T_{ref} pero a la vez, mantener un nivel adecuado para evitar la saturación del controlador. La respuesta del generador en una situación de intervalos nublados es la de suaves escalones de T_{ref} que se realizan si durante un periodo considerable la radiación solar ha variado significativamente. Durante un periodo de tiempo aproximado de 14 minutos (más o menos a

las 14 horas) la irradiancia cae por debajo de 100 $W \cdot m^{-2}$. A este nivel, el campo para automáticamente el seguimiento hasta que la irradiancia no sube por encima de 125 $W \cdot m^{-2}$. Cuando esto ocurre, independientemente del caudal que circule, al no haber energía, la T_{out} va a descender, incluso por debajo de la T_{in} debido a las pérdidas térmicas. En estos casos, los algoritmos tipo IPD tienen problemas de acumulación del error, ya que no disponen de un mecanismo para detectar que T_{ref} no es alcanzable al no haber energía. Lo importante en esta situación, es que, cuando haya suficiente energía como para que el campo solar continúe el seguimiento solar, el GCMB incrementa T_{ref} de forma suave y ayude al algoritmo de control a rebajar el error acumulado. El generador de consignas establece la consigna al mínimo establecido (30 °C) por encima del valor de T_{in} de forma que, al incrementarse la energía solar por el paso del transitorio, el algoritmo de control recupera el seguimiento sin los problemas de oscilaciones que debieran haber derivado de la acumulación del error.

El ensayo mostrado en la figura 8 muestra una fase de arranque donde la dinámica es más rápida y no se producen variaciones importantes en T_{in} . Al final del ensayo, se provocó una perturbación en T_{in} para comprobar la reacción del generador de consignas borroso. Se observa cómo el GCMB modifica el valor de T_{ref} al observar un cambio en T_{in} para mantener los niveles de incremento de temperaturas dentro de los límites de seguridad. A la vez, se da la situación de que la radiación está bajando debido a la caída que se produce en la tarde, por lo que T_{ref} no vuelve a incrementarse y el ΔT se mantiene en niveles inferiores de 80 °C.

En los ensayos mostrados en las figuras 6, 7 y 8 se puede comprobar la actuación de la capa de regu-

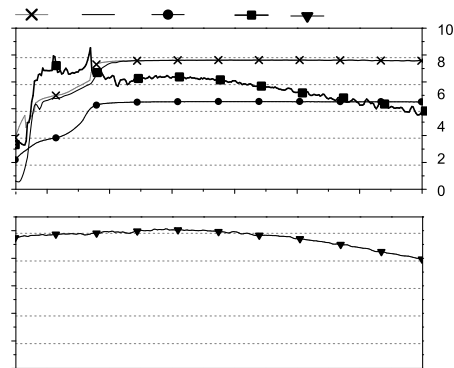


Figura 6: Ensayo completo en la planta ACUREX del GCMB con alta radiación solar

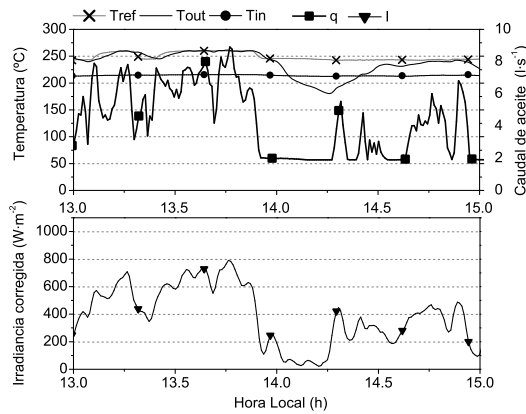


Figura 7: Ensayo completo en la planta ACUREX del GCMB con perturbaciones en la radiación solar

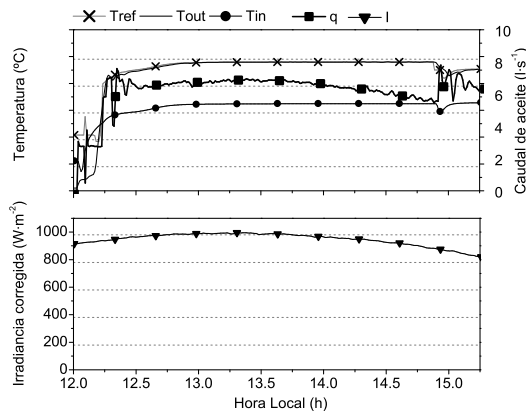


Figura 8: Ensayo completo en la planta ACUREX del GCMB con perturbaciones en la temperatura de entrada al campo

lación (capa 1) de la estructura de control jerárquica mostrada en la figura 3. Cabe destacar el seguimiento conseguido de T_{ref} a pesar de las perturbaciones en la temperatura de entrada (figuras 6 y 8) así como el rechazo de perturbaciones de radiación solar (figura 7). Los ensayos mostrados en este apartado no pretenden evaluar la rapidez de respuesta de la capa de regulación ante un cambio de consigna, pero si el rechazo a las perturbaciones que afectan a la planta así como al correcto seguimiento de la consigna sin error significativo en estado estacionario.

5. CONCLUSIONES

Los resultados mostrados en este trabajo muestran el buen comportamiento de la estructura de control jerárquica formada por un generador de consignas basado en lógica borrosa (capa 2) y una capa de regulación (capa 1) formada por una estructura de control por adelanto y control IPD en paralelo. Mediante ensayos experimentales en la planta real se ha demostrado la utilidad de este tipo de control jerárquico aplicado a plantas solares en distintas condiciones de operación (operación con perturbaciones en la radiación solar, en la temperatura de entrada y en la fase de arranque del sistema). Estas situaciones pueden ser consideradas como difíciles para una operación manual de la planta si se quiere obtener un adecuado rendimiento de la misma alargando el ciclo de vida de los elementos que la forman. Los resultados mostrados prueban que un generador de consignas automático contribuye a facilitar la operación de la planta, simplificando el trabajo del operador. Por otro lado, el GCMB mejora el rendimiento de los controladores evitando su saturación en situaciones en las que la consigna impuesta por el operador es inalcanzable debido a que la cantidad de radiación solar incidente es demasiado baja. La estructura de control implementada es sencilla y responde adecuadamente al rechazo de perturbaciones y al seguimiento de la consigna calculada por la capa 2 del control jerárquico en los distintos puntos de operación en los que ha sido probada.

Agradecimientos

Este trabajo ha sido financiado a través de los fondos CICYT-FEDER (DPI2004-07444-C04-04) y por la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía. Este trabajo ha sido desarrollado en el ámbito del acuerdo específico de colaboración entre la Plataforma Solar de Almería y el Grupo de Investigación “Automática, Electrónica y Robótica” de la Universidad de Almería (TEP197) titulado “Desarrollo de sistemas y herramientas de control para plantas termosolares”.

Referencias

- [1] Bemporad A., Casavola A., Mosca E. (1998). A predictive reference governor for constrained control systems. *Computers in Industry*, 36.
- [2] Berenguel, M., Camacho, E.F., Rubio, F.R. (1993) Simulation software package for the ACUREX field, Internal report ESI Sevilla. <http://www.esi.us.es/~rubio>.
- [3] Berenguel, M., Camacho, E.F., Rubio, F. R., Luk, P.C.K (1997). Incremental Fuzzy PI Control of a Solar Power Plant. In: IEE Proceedings - Control Theory and Applications (Part D)(pp. 596-604), vol. 144, n. 6.
- [4] Berenguel, M., Rubio, F. R., Camacho E.F., Gordillo, F. (1999). Techniques and Applications of Fuzzy Logic Control of Solar Power Plants. In: Fuzzy Theory Systems Techniques and Applications, vol 2 (25). C.T. Leondes: Academic Press.
- [5] Camacho E. F., Berenguel M., Rubio F.R., (1997). Advanced Control of Solar Plants. London: Springer.
- [6] Camacho, E.F., Rubio, F.R., Berenguel, M. (1997). Application of Fuzzy Logic Control to a Solar Power Plant. In Fuzzy Logic Techniques in Power Systems. Ed.: W. Mielczarski. Serie Studies in Fuzziness and Soft Computing. Physica Verlag.
- [7] Cirre M. C., Valenzuela L., Berenguel M., Camacho E.F. (2004). Control de plantas solares con generación automática de consignas. *Revista Iberoamericana de Automática e Informática Industrial* 1 (1), 50-56.
- [8] Duarte O.G. (2004). UNFUZZY-Software, <http://ohm.ingsala.unal.edu.co/ogduarte>.
- [9] Flores, A., Sáez, D., Araya, J., Berenguel, M., Cipriano, A. (2005). Fuzzy predictive control of a solar plant. *IEEE Trans. on Fuzzy Systems*, 13(1), 58-68.
- [10] Johansen T.A., Hunt K.J., Petersen I. (2000). Gain-scheduled control of a solar power plant. *Control Engineering Practice*, 8, 1011-1022.
- [11] Juuso E., Valenzuela L. (2003). Adaptive Intelligent Control of a Solar Collectors Field. In: Wissenschaftsverlag Mainz, Aachen (Ed.), Final Programme and Proceedings of Eunit 2003 - European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (pp. 26-35). Oulu, Finland.
- [12] Kasabov N. K. (1996). Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press.
- [13] Ogunnaike B., Harmon W. (1994). Process Dynamics, Modeling, and Control. New York: Oxford University Press.
- [14] Passino K. M, Yurkovich S. (1998). Fuzzy Control. Menlo Park, California: Addison-Wesley.
- [15] Rubio F., Berenguel M., Camacho E.F. (1995). Fuzzy logic control of a solar power plant. *IEEE Trans. on Fuzzy Systems*, 3 (4), 459-468.

SELECCIÓN DINÁMICA DE MÉTODOS EN LA NAVEGACIÓN DE ROBOTS MÓVILES

Rafael Guirado Clavijo, José Fernando Bienvenido Bárcena, Clara Marcela Miranda Sarmiento
Área de Lenguajes y Sistemas Informáticos. Dpto. de Lenguajes y Computación. Universidad de Almería
rguirado@ual.es, fbienven@ual.es, cmiranda@ual.es

José Carlos Moreno Úbeda
Área de Ingeniería de Sistemas y Automática. Dpto. de Lenguajes y Computación. Universidad de Almería
jcmoreno@ual.es

Resumen

*Desde principios de la década de los 90, se ha venido utilizando la metodología **CommonKADS** en el desarrollo de sistemas basados en conocimiento ubicados dentro de los más diversos ámbitos de lo que ya deberíamos ir acostumbrándonos a llamar **sociedad del conocimiento** (término que, todo sea dicho, ya fue utilizado por Peter Drucker en 1969 [4]).*

*En este trabajo se pretende analizar el efecto que podría tener la aplicación de metodologías de ingeniería del conocimiento, concretamente **CommonKADS** con **selección dinámica de métodos**, en el subsistema de navegación de un robot móvil que se desplaza por un entorno conocido (interior de un invernadero).*

Concretamente, se trata de un robot fumigador que se desplaza por todos y cada uno de los pasillos del invernadero teniendo que cubrir toda la extensión de éste, accediendo a la totalidad de la masa vegetal que contiene.

Palabras Clave: CommonKADS, Sistemas basados en conocimiento, Sociedad del conocimiento, Selección dinámica de métodos, Navegación de robots móviles.

1 INTRODUCCIÓN

La robótica está asumiendo progresivamente un papel fundamental en la realización de tareas que introducen un factor de riesgo humano elevado. En muchos de esos casos, se necesita que el autómata se desplace dentro de un entorno bien definido. De esta forma, continuamente se están diseñando robots móviles para el control de sistemas industriales, para los tratamientos con productos nocivos para la salud humana o para el acceso a lugares donde al hombre

le resultaría muy complicado hacerlo, por citar sólo algunos casos concretos.

En este sentido, y dado que la misión de estos robots es reemplazar a un humano, debe intentar alcanzarse el máximo ajuste posible a su comportamiento. Con este fin, la ingeniería del conocimiento puede ser una de las herramientas que nos proporcione las técnicas para representar el conocimiento y razonamiento humanos dentro del dominio específico, que posteriormente se implantará en el sistema automático desarrollado.

Una de las metodologías para el desarrollo de sistemas basados en conocimiento que más se está utilizando en la actualidad (sobre todo en Europa) es CommonKADS. Así, dentro del marco del proyecto del plan nacional I+D AGL2005-00848, financiado por el Ministerio de Educación y Ciencia, en el que el objetivo final es la construcción de un robot fumigador, se ha incluido una línea en la que se plantea la utilización de la metodología CommonKADS para el desarrollo del sistema de navegación de dicho robot.

El resto del artículo se ha estructurado de la siguiente forma. En la sección 2 se describe la metodología utilizada en el desarrollo, definiendo los conceptos necesarios para su correcta comprensión. En la sección 3 se desarrolla el modelo de conocimiento de la navegación de robots móviles utilizando la metodología CommonKADS con selección dinámica de métodos. Por último, en la sección 4 se muestran las conclusiones del trabajo y la continuación que se le pretende seguir en esta línea.

2 METODOLOGÍA

Como se ha comentado en el apartado anterior, en este trabajo se utiliza la metodología CommonKADS, con la aplicación de selección dinámica de métodos, para la representación del modelo de conocimiento asociado a la navegación de

robots móviles. Además, conviene hacer una breve reseña acerca de los aspectos básicos de la navegación que se utilizan en robótica móvil citando los algoritmos que se han tenido en cuenta al planificar el sistema.

2.1 ALGORITMOS DE NAVEGACIÓN

Existen cuatro partes fundamentales en un robot móvil: el controlador, los sensores, los actuadores y la estructura mecánica. El controlador gobierna todos los movimientos y se compone de tres subsistemas: subsistema de control, subsistema de navegación y subsistema sensorial [7]. El buen funcionamiento del robot dependerá del buen funcionamiento de los tres subsistemas. El subsistema de navegación es el encargado de generar la trayectoria libre de obstáculos óptima para el robot. Esta trayectoria se proporciona como referencia al subsistema de control que genera las consignas a los actuadores teniendo en cuenta también la información proporcionada por el subsistema sensorial.

El problema de navegación se puede descomponer, en el caso más general, en: planificación de tareas, planificación del movimiento, generación de trayectorias y evitación de obstáculos no esperados [6]. En la figura 1 se muestra una arquitectura de control típica de robots móviles.

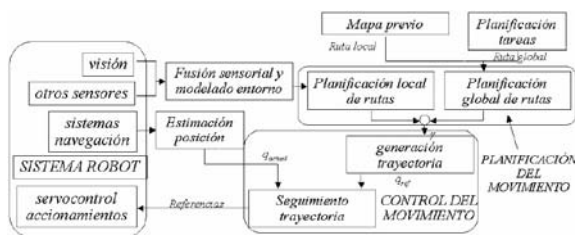


Figura 1: Arquitectura de control típica en robótica móvil

Cada bloque del diagrama de la figura 1 incluye algoritmos especializados para resolver ese problema específico. Este trabajo se centra en la parte de planificación del movimiento.

En un primer nivel de clasificación, se puede hablar de dos enfoques para la planificación del movimiento de un robot móvil: las técnicas globales y las técnicas locales. En las primeras se considera conocido el entorno de movimiento del robot, mientras que en las segundas no es así.

En el caso de aplicación de este trabajo, al tratarse del desplazamiento de un robot por el interior de un invernadero, se podría plantear la utilización del método global, mediante la definición de una especificación de su espacio interior; pero puede darse la circunstancia de que se produzcan

alteraciones en dicha definición (una caja que se queda olvidada en un pasillo del invernadero, una planta que cae obstaculizando el paso, etc.), lo que obliga a combinar los planificadores basados en mapa con otros algoritmos para sortear obstáculos inesperados, empleando así una combinación de técnicas tanto globales como locales en la navegación, que además dota aún de más sentido la búsqueda de una representación de modelo de conocimiento genérico, que incluya tanto algoritmos globales como algoritmos locales.

En la figura 2 se muestran los algoritmos más comunes de planificación del movimiento clasificados en los dos grandes grupos comentados anteriormente: los algoritmos de planificación global y los algoritmos de planificación local.

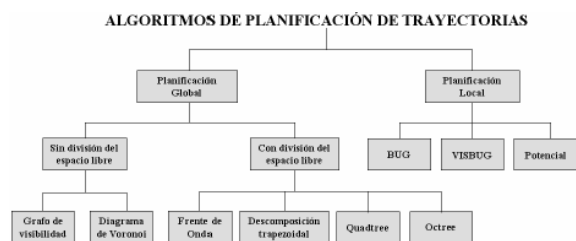


Figura 2: Algoritmos más comunes en la planificación del movimiento

En [5] se presenta una breve descripción de algunos de estos algoritmos junto con una herramienta que permite observar el comportamiento de cada uno de ellos en diferentes situaciones de forma totalmente interactiva, permitiendo así extraer conclusiones acerca de las funciones de coste que se pueden utilizar en un modelo de conocimiento para la selección dinámica de algoritmos de planificación.

Para el propósito de este trabajo el número de algoritmos utilizado carece de importancia, por esta razón se han utilizado únicamente los algoritmos Convex-hull, Grafos-de-visibilidad, Diagramas-de-Voronoi y Campos-potenciales (mostrando la especificación detallada sólo para el primero y el último de ellos) con objeto de presentar la metodología de una forma sencilla. La extensión a un número mayor de algoritmos es directa.

2.2 METODOLOGÍA COMMONKADS

Desde mediados de la década de los 60, se ha intentado modelar el conocimiento para el desarrollo de sistemas informáticos. Así, y muy vinculado al ya consolidado concepto de ingeniería del software, surge el concepto de ingeniería del conocimiento que, en un principio, pretendía dotar de inteligencia (artificial) los sistemas informáticos, dando lugar así a los llamados sistemas expertos o sistemas basados en conocimiento.

La metodología CommonKADS surge como un avance de dicha ingeniería del conocimiento y con el fin de seguir desarrollando sistemas que puedan incorporar conocimiento de forma más sencilla [8].

2.2.1 Definición de CommonKADS

Se trata de una metodología estructurada que soporta todas las fases de desarrollo de un sistema basado en conocimiento, desde el estudio de viabilidad a la implementación, pasando por el análisis y el diseño.

Otro aspecto muy importante de esta metodología es que presenta un soporte para la gestión del conocimiento, por lo que no sólo se centra en la modelización inicial, sino también en su actualización y mantenimiento.

2.2.2 Conjunto de modelos de CommonKADS

Los modelos de CommonKADS permiten describir el conocimiento de la solución de problemas en un dominio particular utilizando distintos niveles de abstracción que posibilitan al ingeniero de conocimiento el detallar el proceso de solución utilizando mecanismos no ligados al dominio y de forma independiente a la implementación. La idea central de la metodología CommonKADS es agrupar los datos relevantes en modelos separados.

En la figura 3 se muestran los diferentes modelos que soportan el análisis del conocimiento en CommonKADS y que constituyen su núcleo; como se puede observar, se estructuran en tres niveles en los que, progresivamente, se pretende responder a cuestiones de contexto (¿por qué?), de concepto (¿qué?) y de realización (¿cómo?).

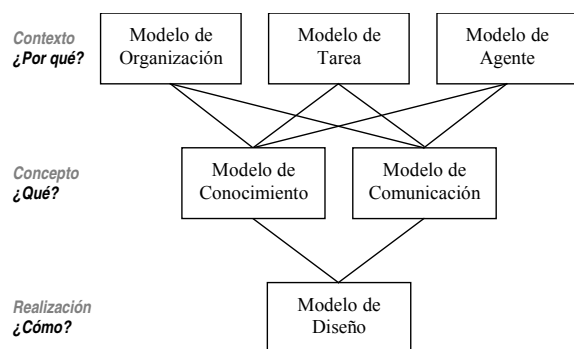


Figura 3: Conjunto de modelos de CommonKADS

2.2.3 Modelo de conocimiento (o experiencia)

De entre los seis modelos mostrados en el apartado anterior, en este trabajo nos centramos exclusivamente en el modelo de conocimiento o experiencia, puesto que describe, de forma detallada e independiente, el conocimiento necesario para

realizar una tarea global (cuyo objetivo es el del sistema). En este sentido, este modelo debe incluir los tipos y estructuras del conocimiento utilizado.

En CommonKADS el conocimiento está diferenciado, dependiendo del tipo de conocimiento que se trate, estableciéndose cuatro grupos: conocimiento del dominio, de inferencias, de tareas y de métodos, como se puede apreciar en la figura 4.

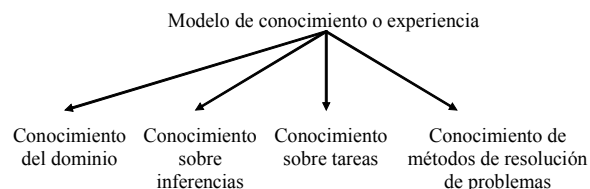


Figura 4: Componentes del modelo de conocimiento

2.2.4 Modelo de tareas y métodos

El diagrama más representativo para modelar mecanismos de solución a problemas (o de consecución de objetivos) dentro del modelo de conocimiento lo constituye el diagrama de tareas y métodos (TMD), que muestra cada una de las tareas que puede realizar un robot móvil así como los métodos que permiten conseguir el/los objetivo/s perseguido/s por cada una de ellas (la figura 5 muestra un ejemplo genérico de este tipo de diagrama, en el que se pueden apreciar las tareas –la global constituye el problema a resolver–, los métodos, las funciones de transferencia y las primitivas o inferencias) [2]. Hay que destacar que, en el campo de la navegación de robots, predominan los métodos que implementan algoritmos en tiempo real y que requieren una ejecución autónoma.

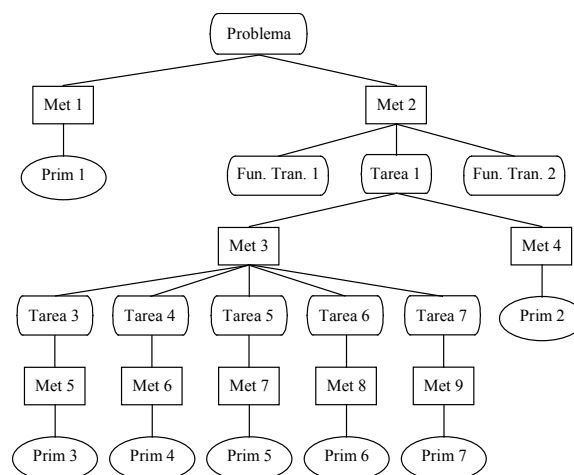


Figura 5: TMD simple

En el ámbito de la robótica móvil, se puede encontrar una gran diversidad de autómatas construidos para conseguir unos objetivos muy distintos. Por ejemplo,

en el terreno militar se han desarrollado robots móviles para la detección de minas, en el terreno agronómico se han desarrollado robots autónomos para la recolección de frutos en invernadero, etc. Pero en cualquier caso, el desplazamiento de un robot móvil siempre se basará en los mismos algoritmos (independientemente del objetivo que persiga).

Así, se puede decir que los algoritmos de navegación de robots móviles son genéricos y, según las necesidades de cada desarrollador, los aplicará de una u otra manera para alcanzar un objetivo determinado.

2.3 SELECCIÓN DINÁMICA DE MÉTODOS

Los métodos que se aplican para la realización de una tarea concreta no tienen por qué ser exclusivos, es decir, puede existir más de un método alternativo para la realización de la misma tarea. En este sentido, y dado que las condiciones del entorno (y del propio robot) suelen ser cambiantes, puede resultar necesario que el sistema tenga que decantarse “sobre la marcha” por la aplicación de un método alternativo concreto de entre varios posibles.

Clásicamente, este proceso consistiría en construir un módulo de selección que, teniendo en cuenta límites bien definidos, activaría el método (proceso) más adecuado. En nuestro caso, se propone un gestor de opciones independiente de éstas, que a partir de las condiciones externas e internas, opte por activar el método más adecuado. La adecuación de los métodos se evalúa según un conjunto de criterios de aptitud cuyo valor y peso pueden ser actualizados tanto manual como automáticamente [3].

Se plantean distintos esquemas generales de selección, como el que se muestra en la figura 6, que representa el diagrama TMD extendido de la tarea general selección-en-cascada.

Para la toma de decisiones, el sistema debe basarse en las condiciones actuales del entorno y en las posibilidades (características) del propio robot. Por ejemplo, un robot que controla que la temperatura de diversas cubetas de agua no supere un determinado umbral, para lo que va introduciendo su brazo secuencialmente en cada una de ellas, tiene una pieza plástica en su brazo que se derrite si el agua alcanza su punto de ebullición; en este caso, el robot debe estar preparado para no introducir su brazo en una cubeta que alcance su punto de ebullición (condiciones del entorno) si no se quiere que sufra un daño (características del robot). Aunque puede considerarse ficción (o quizás un adelanto de la realidad futura), conviene recordar en este momento la tercera ley de la robótica de Isaac Asimov: “Un robot debe proteger su propia existencia en la medida

en que esta protección no entre en conflicto con la Primera o la Segunda Ley” [1].

Dado que se pretende realizar un modelado genérico del conocimiento, habrá que considerar situaciones que no sean aplicables a todos los robots móviles. Por ejemplo, si un algoritmo concreto precisa obligatoriamente que el robot disponga de un sensor de ultrasonidos, sólo podrá aplicarse dicho algoritmo si se ha cumplido tal condición en la construcción del mismo.

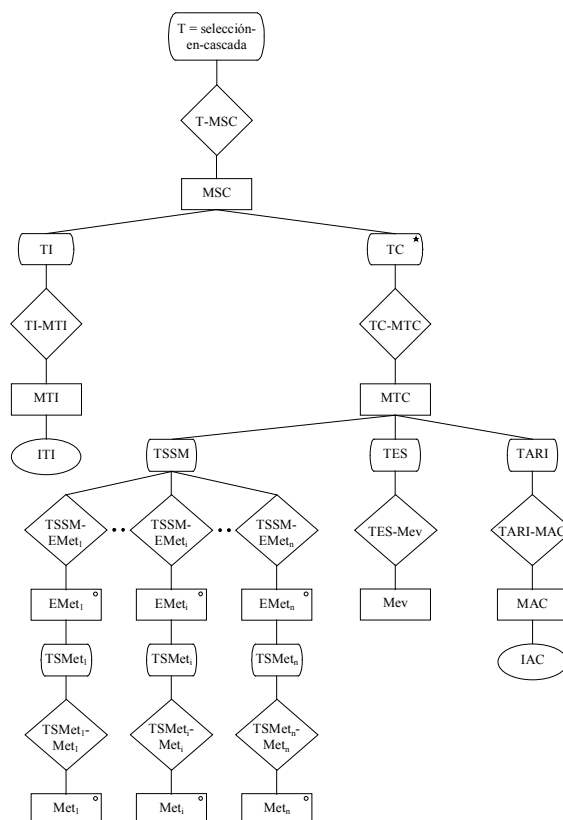


Figura 6: Diagrama TMD de la tarea selección-en-cascada

3 DESARROLLO DEL MODELO

Con el modelado del conocimiento se pretende hacer explícito todo el conocimiento implícito en los textos redactados por los expertos en la materia en cuestión (en nuestro caso, la navegación de robots móviles). Esta es la razón por la que este tipo de tareas requieren una consulta bibliográfica enormemente amplia y evolutiva.

3.1 MODELO DE CONOCIMIENTO

Se está desarrollando un modelo de conocimiento genérico, según la metodología CommonKADS con selección dinámica de métodos, del subsistema de

navegación de un robot móvil. Dentro de este ámbito, podemos encontrarnos con una amplísima casuística (características del entorno y de los obstáculos que presenta, características cinemáticas y dinámicas del robot, sensores utilizados, etc.). Por poner un ejemplo de caso extremo, podemos tener diversos tipos de obstáculos dentro del mismo entorno, de forma que unos sean más “peligrosos” para el robot que otros (por ejemplo, en el sentido de que sean menos estables y puedan caer con facilidad sobre el robot, si éste entra en contacto con ellos, causándole algún daño). Por este motivo, el modelo de conocimiento debe incluir una especificación muy detallada de cada uno de los elementos que intervienen en la tarea de navegación con el fin de que el robot seleccione la alternativa más adecuada a cada situación particular.

En este modelo de conocimiento, se parte de un problema global que consiste en conseguir que un robot navegue correctamente por un entorno para alcanzar un objetivo determinado (esta sería, por tanto, la tarea principal de nuestro modelo). Para conseguir el objetivo de esta tarea, según las condiciones del sistema (en este caso, que se

disponga o no de un mapa del entorno, es decir, que éste sea o no conocido), se pueden plantear dos posibles métodos iniciales: global y local. El primero de los métodos plantea el objetivo de calcular una trayectoria eficiente según algún criterio preestablecido (distancia mínima, mayor alejamiento posible de los obstáculos, ...), por lo que nos surge una nueva tarea. El segundo método puede requerir la aplicación de una tarea de exploración (reconocimiento del entorno para generar un mapa del mismo) o la aplicación de una tarea que comience la navegación del robot en busca de su objetivo detectando y evitando los obstáculos sobre la marcha. Si se realiza una exploración del entorno generando un mapa del mismo, el método a aplicar sería el global (comentado anteriormente) puesto que tendríamos un nuevo estado en las condiciones del problema idéntico al que nos decantaba por el primer método en el primer planteamiento de alternativas.

Como el desarrollo de este modelo de conocimiento completo sería excesivamente grande, en la figura 7 nos limitamos a mostrar como ejemplo sólo algunas de las ramas del TMD correspondiente.

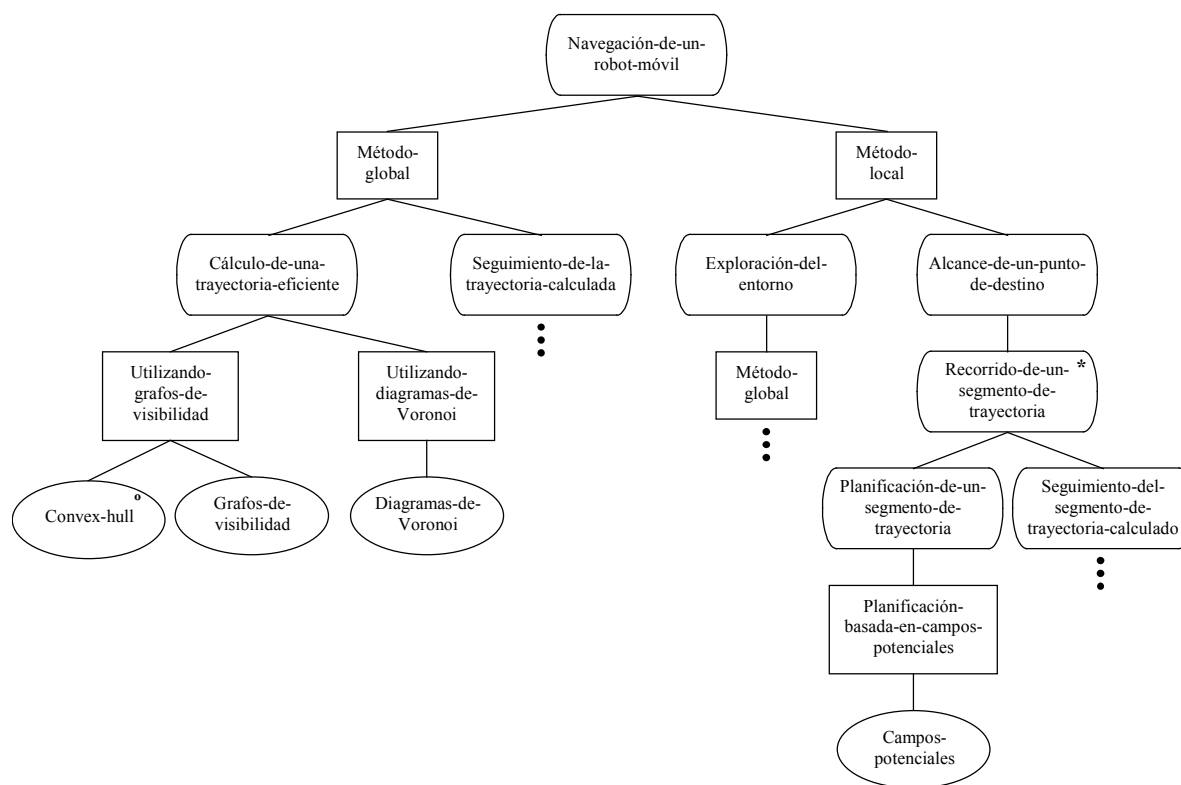


Figura 7: Representación parcial del diagrama TMD para la navegación de un robot móvil

La tarea Recorrido-de-un-segmento-de-trayectoria repite iterativamente las dos sub tareas en que se divide, Planificación-de-un-segmento-de-trayectoria y Seguimiento-del-segmento-de-trayectoria-calculado, estrictamente en ese orden, hasta alcanzar el destino especificado o hasta comprobar que es

inalcanzable. Esto se ha representado en el diagrama mediante un * en la parte superior derecha del símbolo que representa a la tarea.

La inferencia Convex-hull, que calcula la envolvente convexa de cada obstáculo o conjunto de obstáculos,

sólo se lanzará cuando exista algún obstáculo o conjunto de obstáculos muy próximos que presenten algún vértice cóncavo en su polígono envolvente. Como se comenta en el esquema CML correspondiente (apartado siguiente), tampoco se lanzará esta inferencia si la posición de origen o de destino del robot se encuentra dentro de la parte cóncava de un obstáculo (como mínimo debe dejar de aplicarse sobre el/los vértice/s cóncavo/s sobre los que se encuentra la configuración inicial o la final con el fin de que no pase a estar contenida en el obstáculo).

La selección dinámica de métodos se aplica, por ejemplo, en la tarea de cálculo de una trayectoria eficiente. Se han planteado dos alternativas (aunque existen muchísimas más) para la realización del objetivo perseguido por dicha tarea, cada una de las cuales tiene definidos unos criterios de aplicabilidad (cada uno con su peso) que permitirán al sistema tomar la decisión más apropiada ante unas determinadas condiciones del sistema.

3.2 DESARROLLO DE LAS INFERENCIAS / PROCESOS

Para especificar cada uno de los componentes del TMD anterior (tareas, métodos e inferencias), CommonKADS plantea la utilización de esquemas CML. Estos esquemas muestran todo el conocimiento involucrado en cada elemento. A continuación se muestran las partes relevantes de la sintaxis de alguna de las inferencias del diagrama, que representan los procesos que se ejecutan para la consecución, en primera instancia, de los objetivos parciales del sistema, para terminar alcanzando el objetivo de la tarea principal.

La descripción simplificada de la inferencia Convex-hull sería la siguiente:

INFERENCE Convex-hull;

OPERATION-TYPE: calcular;

ROLES:

INPUT:

grafo-visibilidad:

“descripción formal del grafo constituido por todos los vértices de los obstáculos del entorno y las configuraciones inicial y final; estos vértices se emparejan según su visibilidad, es decir, si la línea que los une no intersecta con ningún obstáculo”;

OUTPUT:

grafo-visibilidad-convexo:

“descripción formal del grafo de visibilidad considerando los obstáculos como polígonos convexos; si la configuración inicial o final está dentro de la parte cóncava de un obstáculo, no se debe aplicar esta inferencia”;

SPECIFICATION:

“1. Buscar vértices con valores máximos y mínimos de x e y (4 vértices).

2. Ordenar todos los vértices en orden ascendente de x .

3. Empezar con el vértice con menor x (x_{\min}). Ponerlo sobre una pila y buscar extremos entre vértice x_{\min} y vértice y_{\min} .

a. Calcular la pendiente de todas las líneas que parten del vértice que se encuentra en lo alto de la pila y va hacia todos los vértices que se encuentran a la izquierda del correspondiente a y_{\min} y a la derecha y debajo del que se encuentra en lo alto de la pila.

b. Seleccionar el vértice con el que se haya trazado la línea con pendiente más negativa y ponerlo sobre la pila.

4. Repetir paso 3 con el nuevo vértice sobre la pila hasta que el vértice correspondiente a y_{\min} se ponga en la pila.

5. Repetir pasos 3 y 4 trazando las líneas desde el vértice correspondiente a y_{\min} y el de x_{\max} , buscando en vértices a la derecha del de y_{\min} y a la izquierda y debajo del de x_{\max} con la pendiente positiva menor.

6. Repetir pasos 3 y 4 trazando las líneas desde el vértice correspondiente a x_{\max} y el de y_{\max} , buscando en vértices encima de x_{\max} y a la derecha y debajo del de y_{\max} con la pendiente negativa mayor.

7. Repetir pasos 3 y 4 trazando las líneas desde el vértice correspondiente a y_{\max} y el de x_{\min} , buscando en vértices debajo de y_{\max} y a la derecha y encima del de x_{\min} con la pendiente positiva menor”

END INFERENCE Convex-hull;

Para la inferencia Campos-potenciales, ésta sería su descripción simplificada:

INFERENCE Campos-potenciales;

OPERATION-TYPE: calcular;

ROLES:

INPUT:

configuración-de-destino:

“coordenadas del punto que el robot pretende alcanzar”;

información-sensores:

“datos procedentes de los sensores del robot (ultrasonidos, visión, ...) y que informan sobre la ubicación de los obstáculos que se encuentran en la trayectoria que debería seguir el robot para alcanzar el punto objetivo”;

definición-potenciales:

“definición de los potenciales de atracción y repulsión; el de atracción debe ir en función de la distancia euclídea a la posición destino (cuanto más cerca esté el robot, menor será su

influencia); el potencial repulsivo conviene que sólo influya en el movimiento del robot cuando éste se encuentre demasiado próximo a un obstáculo, de tal forma que la fuerza debida a este hecho tenga una dirección que lo aleje del mismo”;

OUTPUT:

vector-fuerza-artificial:

“vector fuerza artificial $F(p)$ que especifica el sentido, la dirección y la aceleración que debe seguir el robot en su desplazamiento; se utiliza como entrada de la tarea Seguimiento-del-segumento-trayectoria-calculado, cuyo objetivo es el desplazamiento del robot en función de este vector calculado”;

SPECIFICATION:

“1. Calcular el potencial $U(p)$ que actúa sobre el robot en la posición actual p según la información recabada de los sensores.

2. Determinar el vector fuerza artificial $F(p)$ mediante la expresión $F(p) = -\nabla U(p)$ ”

END INFERENCE Campos-potenciales;

De forma similar, se representarían los esquemas CML del resto de elementos (tareas, métodos, funciones de transferencia e inferencias) que constituyen el diagrama TMD del modelo de conocimiento.

4 CONCLUSIONES Y TRABAJOS FUTUROS

La metodología CommonKADS nos proporciona las herramientas necesarias para modelar el conocimiento implícito en cualquier entorno sobre el que se pretenda construir un sistema basado en conocimiento (o simplemente sistema de conocimiento). En este sentido, y aunque en este trabajo nuestro campo de aplicación sea muy concreto, la idea es que se tengan muy claros los métodos que se deben aplicar para resolver un problema concreto.

En última instancia, se pretende mejorar la fase de implementación de la metodología incorporando una herramienta que, a partir del diseño, genere un sistema software que se limite a lanzar el proceso que se necesite en cada momento (según el objetivo que se persiga).

Agradecimientos

Este trabajo no habría sido posible sin la inestimable colaboración de todo el resto de miembros del área de Ingeniería de Sistemas y Automática de nuestro departamento, entre los que se encuentran Manuel Berenguel, Francisco Rodríguez y José Luis Guzmán.

Referencias

- [1] Asimov, I. (1942) “Runaround”, *Astounding Science Fiction*. Reeditado en Robot Visions, Isaac Asimov, ROC, 1990.
- [2] Bienvenido, J.F. (1999) “Selección dinámica de métodos y reutilización de elementos de conocimiento: Una extensión a CommonKADS aplicada a problemas de diseño”, *Tesis Doctoral*, Universidad de Murcia.
- [3] Bienvenido, F.; Flores-Parra, I.; Guirado, R.; Marín, R. (2001) “Knowledge Based Modeling of the Design Process as a Base of Design Tools. Application to the Development of Agricultural Structures”. *Computer Aided Systems Theory EUROCAST'2001*, Lecture Notes in Computer Science, Vol. 2178, pp. 209-222, Springer-Verlag, Heidelberg, Germany.
- [4] Drucker, P. (1969) “The Age of Discontinuity: Guidelines in Our Changing Society”, *Harper and Row*, New York.
- [5] Guzmán, J.L.; López, O.; Berenguel, M.; Rodríguez, F.; Dormido, S.; Pigué, Y. (2004) “MRIT: Mobile Robotics Interactive Tool”, *Internet Based Control Education IBCE'04*, Grenoble (Francia).
- [6] Meyer, J.A.; Filliat, D. (2003) “Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies”, *Journal of Cognitive Systems Research* 4, pp. 283-317.
- [7] Ollero, A. (2001) “Robótica. Manipuladores y robots móviles”, *Marcombo, S.A.* Barcelona, España.
- [8] Schreiber, G.; Akkermans, H.; Anjewierden, A.; de Hoog, R.; Shadbolt, N.; Van de Velde, W.; Wielinga, B. (1999) “Knowledge Engineering and Management. The CommonKADS Methodology”, *The MIT Press*.

ARQUITECTURA DE TELEOPERACIÓN DISTRIBUIDA PARA EQUIPOS DE ROBOTS MÓVILES

G. Cano, R. Ñeco, O. Reinoso, L. Payá, C. Fernández
Dpto. Ingeniería de Sistemas Industriales
Universidad Miguel Hernández de Elche
Avda. Universidad s/n
o.reinoso@umh.es

Resumen

En este artículo se presenta una arquitectura de teleoperación totalmente distribuida para el manejo de un conjunto de equipos de robots móviles cada uno de los cuales se desenvuelve en un entorno remoto diferente. La característica principal de la arquitectura propuesta es su carácter abierto y flexible estando constituida por un conjunto de subsistemas completamente distribuidos. Además, otra de las características principales de la arquitectura propuesta es la posibilidad de admitir una interfaz de comandos de alto nivel con procesamiento de lenguaje natural o interfaz amigable. El número de usuarios que pueden conectarse a esta arquitectura es también múltiple, siendo diversas las posibilidades de conexión. Tan solo es necesario que el cliente disponga de un navegador web compatible con HTML.

Palabras clave: Teleoperación, robots cooperativos, arquitectura distribuida, reconocimiento lenguaje natural.

Los retardos en la comunicación al emplear Internet como medio de transmisión de la información provocan en numerosas ocasiones distorsiones en el control y supervisión de equipos a distancia que es necesario tener en cuenta. En este sentido en [13] se propone una estructura de control estable para teleoperación de robots móviles a través de Internet. En esta estructura se hace uso de una compensación de los retardos en la comunicación tanto en el entorno local como en el entorno remoto. También en [7] se hace uso de una aproximación biológica para la consideración de los retardos existentes en la comunicación y paliar los inconvenientes que se pudieran ocasionar en el control de los robots en el entorno remoto.

En este artículo se presenta una arquitectura de teleoperación distribuida constituida por un conjunto de capas que posibilita el control y supervisión de un conjunto de equipos de robots móviles cada uno de los cuales se desenvuelve en un entorno diferente. Además es posible interactuar con esta arquitectura a través de una interfaz amigable al utilizar comandos de alto nivel en lenguaje natural.

1. INTRODUCCIÓN

La teleoperación de sistemas constituidos por robots móviles ha sido un tema bastante estudiado en los últimos años por diferentes investigadores de todo el mundo. El control y manejo de un conjunto de robots a través de internet presenta considerables dificultades que es necesario tratar y evaluar. Recientemente se han presentado numerosos desarrollos para el manejo y control de robots móviles. Desde aquellos que realizan el control de un robot móvil, como [11] [12], hasta los que consideran un conjunto de robots en los entornos de teleoperación como en [5] y [14]. En el manejo y control de estos robots a través de Internet surgen ciertos problemas como la pérdida de datos en la comunicación que se pueden reducir gracias a mecanismos fiables de comunicación tales como TCP/IP aún a costa de incrementar el retardo en la comunicación. No obstante este problema puede ser solventado al incrementar la capacidad de inteligencia de la aplicación.

1.1. ARQUITECTURAS DE TELEOPERACIÓN

Otro de los temas de interés en los problemas de teleoperación se presenta en las arquitecturas empleadas. Hasta la actualidad se han empleado múltiples esquemas y arquitecturas de control. Por ejemplo, en [9] se presenta un esquema de arquitectura basada en dos partes diferenciadas: una de percepción y otra de acción. Cada una de estas partes se divide en diferentes niveles dependiendo de la acción a realizar. En [1] se presenta una arquitectura tolerante a fallos para el manejo de robots móviles a través de Internet haciendo uso de la programación basada en agentes. Otras arquitecturas tolerantes a fallos se han presentado con anterioridad tanto para el manejo de un único robot [4] como para múltiples robots [10]. Sin embargo, estos sistemas emplean tanto hardware como software de forma redundante con objeto de minimizar los fallos que se pueden generar en el desarrollo de la ejecución. En [6] se presenta

una arquitectura basada en tres capas: una capa cliente en el entorno local y dos capas en el entorno remoto, el servidor y una capa *middleware* que conecta los distintos comportamientos de la arquitectura.

1.2. INTERFACES AMIGABLES

Otro de los elementos fundamentales que se presentan en el desarrollo de sistemas de teleoperación versátiles y flexibles es la posibilidad de disponer de interfaces de comunicación robustas y con un grado de inteligencia que permitan al usuario tener un conocimiento lo más amplio posible de lo que sucede en los entornos remotos. Así por ejemplo, en [8] se presenta un sistema de control de robots (en este caso articulados) a través de internet mediante dispositivos móviles como interfaces de usuario, y el empleo de comandos de voz para el control del dispositivo remoto.

El objetivo principal desarrollado en el presente artículo consiste en presentar una arquitectura de teleoperación diseñada con objeto de manipular y controlar a varios niveles las acciones realizadas por uno o un conjunto de robots que se encuentran en un entorno remoto. La característica principal de la arquitectura desarrollada consiste en posibilitar la ejecución simultánea de múltiples clientes cada uno de ellos a través de una interfaz determinada para la teleoperación conjunta de conjuntos de robots, pudiendo estar cada robot en un entorno remoto diferente. El conjunto o equipo de robots puede estar constituido tanto por un conjunto físico de robots reales como por un sistema de simulación que admita las acciones emitidas por el operador en el entorno local. Otra de las características principales de la aplicación desarrollada consiste en el diseño distribuido de la misma. Cada una de las capas en las que se ha estructurado la arquitectura puede implementarse sobre un sistema diferente lo que le confiere un alto grado de distribución y de flexibilidad.

1.3. RECONOCIMIENTO DE LENGUAJE NATURAL

El reconocimiento de lenguaje natural puede proporcionar al usuario de un sistema teleoperado la posibilidad de usar un interfaz muy cómodo e intuitivo para la ejecución de tareas. Por este motivo, la arquitectura distribuida presentada aquí se ha implementado junto con un sistema de reconocimiento de comandos expresados en lenguaje natural que pueden comandar un conjunto de varios robots.

Este módulo es flexible y adaptable en dos sentidos. Por un lado, es capaz de adaptarse a nuevas

estructuras sintácticas de comandos ya existentes o bien es capaz de crear comandos nuevos a partir de las instrucciones del usuario. Estos nuevos comandos se añaden haciendo uso de una base de datos léxica, como se describe posteriormente. Por otro lado, el sistema teleoperado es capaz de recuperarse de errores de ejecución y aprovechar esta información para mejorar futuras ejecuciones de los mismos comandos.

En trabajos previos se desarrolló un sistema de reconocimiento de lenguaje natural para el control de robots a través de Internet ([2, 3]). En la aplicación presentada en este artículo se realiza la extensión de estos trabajos al entorno constituido por un conjunto múltiple tanto de usuarios como de entornos remotos constituidos por un diferentes equipos de robots.

El resto del artículo queda organizado como sigue. En la siguiente sección se presenta el modelo de arquitectura diseñado de forma distribuida para posibilitar una comunicación fluida entre cualquiera de los clientes conectados al sistema y cualquiera de los posibles entornos de equipos de robots móviles a teleoperar. Dentro de este apartado se analizan con detalle cada una de las capas en las que se ha configurado la arquitectura. En la siguiente sección se presenta el flujo de control típico que se lleva a cabo para ejecutar las acciones emitidas por el usuario. Posteriormente se presentan las interfaces de usuario desarrolladas así como las principales características de las mismas. Por último se presentan las conclusiones del trabajo.

2. MODELO DE ARQUITECTURA

La arquitectura diseñada se encuentra dividida en cinco capas pudiendo estar estructurada a su vez por un conjunto de subsistemas independientes y fuertemente conexiónados. A diferencia de las arquitecturas clásicas de teleoperación en el que el usuario se conecta de forma remota a través de Internet con el proveedor de servicios que establece la comunicación y control a bajo nivel con el entorno remoto y los dispositivos físicos presentes en el mismo, en la arquitectura propuesta todas las capas que comprenden el modelo se encuentran totalmente distribuidas dentro de la red (figura 1).

Así pues la característica fundamental de la arquitectura presentada radica en que cada una de las capas puede ejecutarse en un equipo distinto lo que le confiere un alto grado de paralelismo, flexibilidad y robustez configurando una propuesta altamente distribuida. De esta forma puede considerarse que el conjunto está formado por subsistemas, comunicándose entre ellos mediante el envío

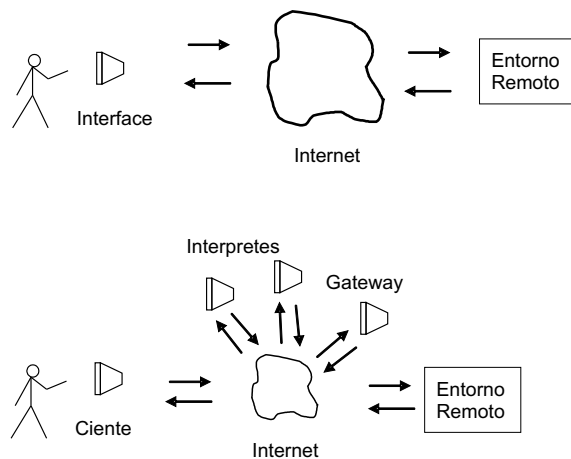


Figura 1: Esquema clásico de teleoperación y esquema propuesto con una solución distribuida

de paquetes IP, formando una pila de protocolos. Las cinco capas en las que se encuentra dividida la arquitectura propuesta son las siguientes (figura 2):

- Clientes
- Interfaces
- Interprete de comandos
- Gateway de control
- Entorno remoto

En los apartados siguientes se introducirán algunas características de cada una de estas capas.

2.1. CLIENTES

Los clientes proporcionan el medio para poder acceder a la aplicación. La característica fundamental de esta capa consiste en que cada usuario puede acceder al sistema utilizando el hardware más variopinto o heterogéneo. Tan solo es indispensable que disponga de un navegador Web compatible con HTML.

De esta forma se consigue homogeneizar el acceso tanto desde un computador personal, como desde un dispositivo móvil PDA, o cualquier otro dispositivo con navegador Web.

2.2. INTERFACES

El interfaz es el método que facilita la interacción del usuario con la aplicación a través del uso de un conjunto de herramientas tanto visuales, textuales, como de análisis de voz. La forma de interaccionar de los clientes a través de la interfaz

puede ser diversa y heterogénea. En la arquitectura implementada es posible interaccionar con los equipos remotos mediante múltiples vías:

- Comandos visuales
- Comandos textuales
- Reconocimiento de lenguaje natural

El interfaz de usuario que se ha implementado en la aplicación se encuentra compuesto por dos elementos fundamentales: por una parte un servidor de Web de XHTML, que genera las páginas HTML que los navegadores de los clientes visualizarán, y por otro lado un Visor de Incidencias (LOGGER) que posibilita la realimentación hacia el usuario de la información útil que se haya generado en el entorno remoto como consecuencia de la aplicación de un comando. El sistema visor de incidencias además de generar las respuestas de la acción mediante comandos textuales, comunica éstas mediante un sistema de síntesis de voz.

2.3. INTÉRPRETE DE COMANDOS

El intérprete de comandos es la capa más compleja de la arquitectura desarrollada y realiza dos tareas básicas:

- El análisis y posterior traducción de la orden dada por el usuario a los robots reales o simulados.
- Se adapta automáticamente a medida que se ejecuta y se desarrolla la aplicación.

El objetivo fundamental de esta capa consiste en validar los comandos que han sido introducidos por los usuarios a través de la capa cliente y determinar si las acciones solicitadas sobre el entorno remoto forman parte de un *lenguaje* que es capaz de reconocer.

Este subsistema está integrado por un conjunto de procesos muy relacionados entre sí que hacen uso intensivo del Diccionario de la Base de Datos donde se almacenan las reglas y palabras que pertenecen a este *lenguaje*.

El análisis que se lleva a cabo con cada una de las órdenes emitidas por el usuario dentro de este intérprete de comandos son:

Análisis léxico: se realiza una verificación de que los caracteres y palabras emitidos por el usuario son permitidos por el lenguaje de comandos. Las acciones están compuestas por

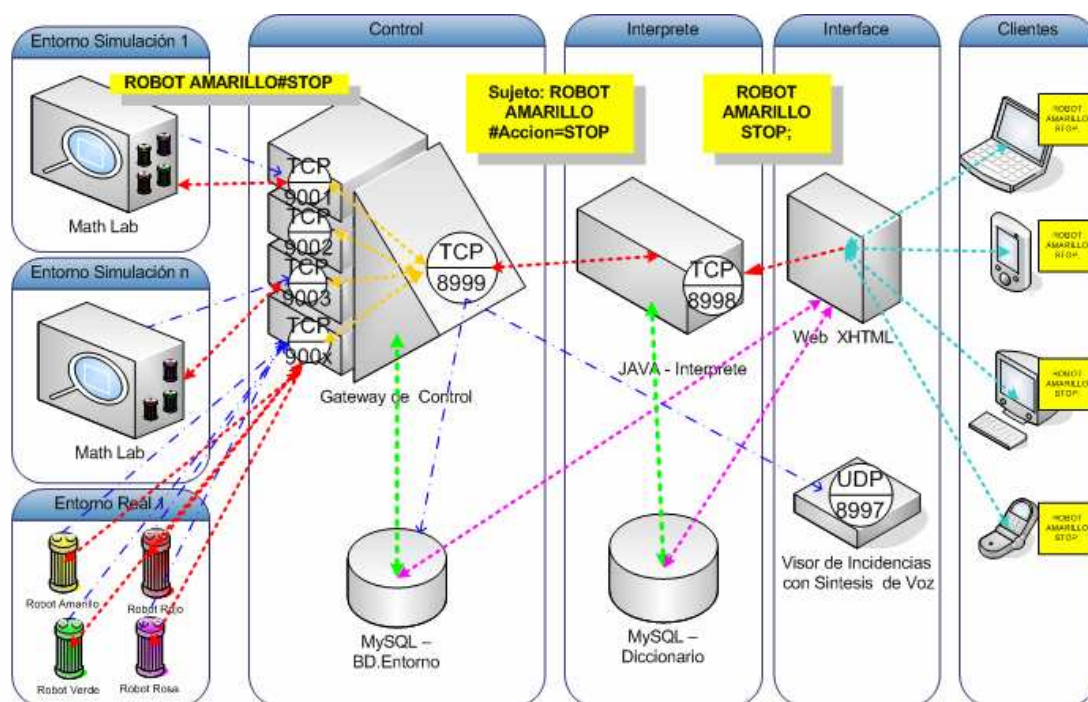


Figura 2: Arquitectura distribuida para la teleoperación del conjunto de equipos de robots

palabras del lenguaje de comandos a reconocer (*tokens* o terminales) que deben pertenecer al diccionario que se encuentra en la base de datos.

Análisis sintáctico: además de que las palabras se encuentren en el diccionario, deben encontrarse en un cierto orden preestablecido en las reglas de producción que determinan la forma en que se deben estructurar o describir las acciones que deben ejecutar los equipos de robots en el entorno remoto.

Análisis semántico: en esta fase se realiza la traducción de los comandos al lenguaje de programación entendible por el sistema a bajo nivel.

La principal ventaja de tener asociada una base de datos radica en que por sus características dinámicas y de flexibilidad, el diccionario puede incrementarse a medida que el sistema evoluciona con el tiempo. De esta manera se dota al sistema con la posibilidad de aprendizaje a medida que se realizan nuevas acciones.

2.4. GATEWAY DE CONTROL

El Gateway de control es el subsistema que realiza la función de enlace entre las acciones solicitadas por el intérprete, y los entornos remotos ya sean virtuales/simulados o físicos/reales. De forma sim-

plificada el objetivo del Gateway consiste en realizar el envío de las acciones comandadas por el usuario al robot dentro del sistema remoto, una vez que éstas han sido interpretadas. De esta forma es pues necesario llevar a cabo un proceso de distribución por el puerto asociado a cada robot en función del entorno de simulación o físico al que pertenece el robot que debe acometer la acción.

Para poder realizar ésta y otras tareas encomendadas, este subsistema se encuentra compuesto de dos elementos: El programa *encaminador de mensajes* y el programa que da acceso a la Base de Datos de Entorno. De esta forma es posible identificar el destino de los mensajes de los robots, puesto que dentro de esta base de datos se almacena toda la información relacionada con los diferentes entornos remotos presentes en la aplicación así como las características e identificación de los robots dentro de cada uno de estos entornos.

Asimismo, el Gateway de control es el subsistema encargado de resolver todas las relaciones geométricas necesarias para poder acometer la acción comandada por el usuario. Un ejemplo de este tipo de relaciones geométricas que se resuelven dentro de este subsistema son las diferentes etiquetas que representan en cada uno de los posibles entornos remotos una posición concreta de la escena (por ejemplo, '*Robot amarillo avanza hasta ZonaCero*').

2.5. ENTORNO REMOTO: ELEMENTOS TERMINALES

Finalmente la última capa se encuentra configurada por cada uno de los posibles entornos remotos en los que operan un conjunto de robots para realizar una determinada tarea. Estos elementos terminales pueden estar constituidos por entornos remotos virtuales o de simulación dentro de los cuales se encuentran realizando diversas tareas los robots asignados o bien por entornos remotos reales en los que se puede asignar diferentes tareas a cada robot dentro de éstos.

3. FLUJO DE CONTROL

El flujo de control determina el camino seguido por las acciones emitidas por el usuario cuando solicitan una determinada actuación sobre un entorno remoto hasta que ésta se lleva a cabo. En función del éxito de la acción solicitada se pueden presentar dos esquemas:

1. Acción reconocida como válida y transmitida para su realización (figura 3). En este caso se realizan las siguientes actuaciones:
 - El usuario demanda una acción de alguno de los entornos remotos.
 - El interfaz envía un mensaje al intérprete como secuencia de esta petición.
 - El intérprete reconoce y valida la acción. En caso positivo la envía al Gateway de control.
 - El Gateway de control encamina el mensaje al robot correspondiente dentro del entorno adecuado para que realice la acción.
 - Se ejecuta la acción solicitada.
 - Eventualmente el robot puede informar de un determinado evento mediante un mensaje enviado al Visor.
2. Acción NO reconocida como válida. En consecuencia se devuelve un mensaje de error (figura 4). En este caso se realizan las siguientes actuaciones:
 - El usuario demanda una acción de alguno de los entornos remotos.
 - El interfaz envía un mensaje al intérprete como consecuencia de esta petición.
 - El intérprete detecta algún error en la decodificación de la información solicitada. En consecuencia NO valida la acción generando un evento de error
 - El usuario es informado de la existencia de tal error.

4. INTERFAZ DE USUARIO

La generación del interfaz de usuario está basada en un modelo Web, que requiere soporte HTML Standard de los navegadores que lo utilicen. Se ha empleado un servidor Apache y extensiones PHP para simplificar la creación de páginas dinámicas, sobre todo para hacer uso de la base de datos y acceso a MySQL que lo simplifican en gran medida. La especificación real es XHTML y CSS, para facilitar la correcta visualización con independencia del hardware del cliente empleado.

El interfaz lo podemos dividir en dos grandes secciones, según sus funciones, que se describen en las dos secciones siguientes.

Interfaz de control

Éste es el punto básico de la aplicación donde se lanzan las acciones que ponen en marcha el sistema. Desde esta interfaz desarrollada (figura 5) se permite la modificación de las acciones solicitadas al entorno remoto, o la introducción de las mismas en un lenguaje de alto nivel.

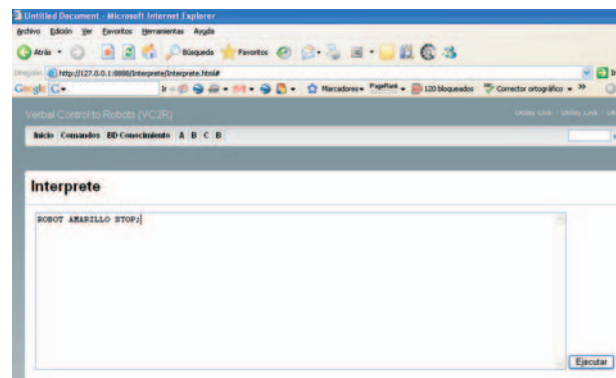


Figura 5: Interfaz de Control

Interfaz de información

Este subsistema ofrece la supervisión de las acciones o actividades solicitadas así como un punto donde los usuarios pueden descubrir cómo se han llevado las acciones, qué acciones reconoce el sistema y por qué. Para llevar a cabo estas funcionalidades, se ha usado el servidor HTTP Apache y el lenguaje PHP para la creación del contenido dinámico (ver figura 6).

Visor de incidencias

El visor de incidencias conforma el elemento funcional capaz de satisfacer la necesidad de transmitir mensajes asíncronos producidos espontáneamente, fruto de acciones de control tomadas en el pasado o como consecuencia de la evolución del sistema en el tiempo.

Se ha escogido la opción de realizar un modelo

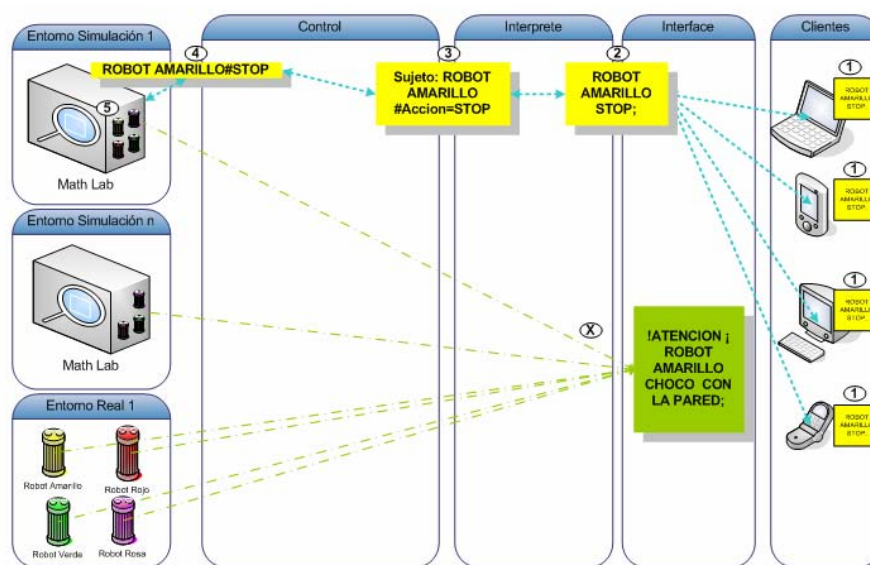


Figura 3: Flujo de control en una acción validada de forma correcta

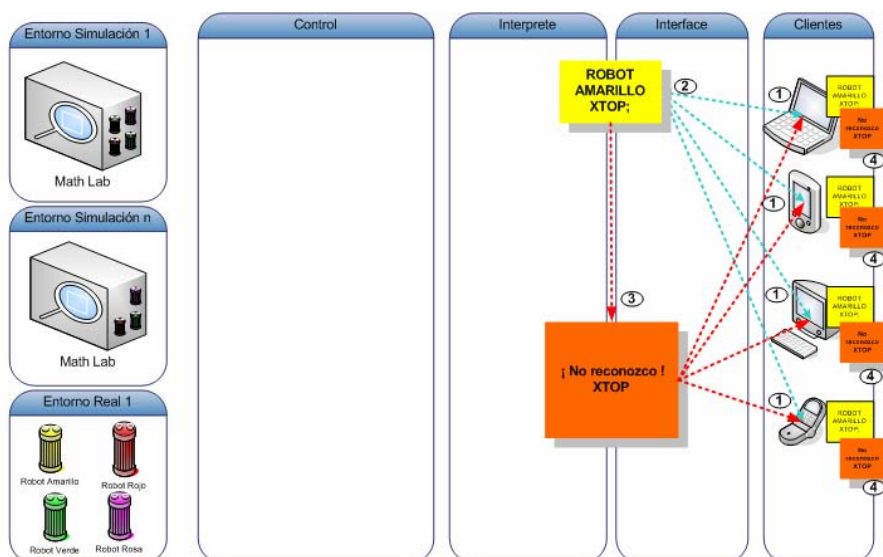


Figura 4: Flujo de control en una acción NO validada de forma correcta

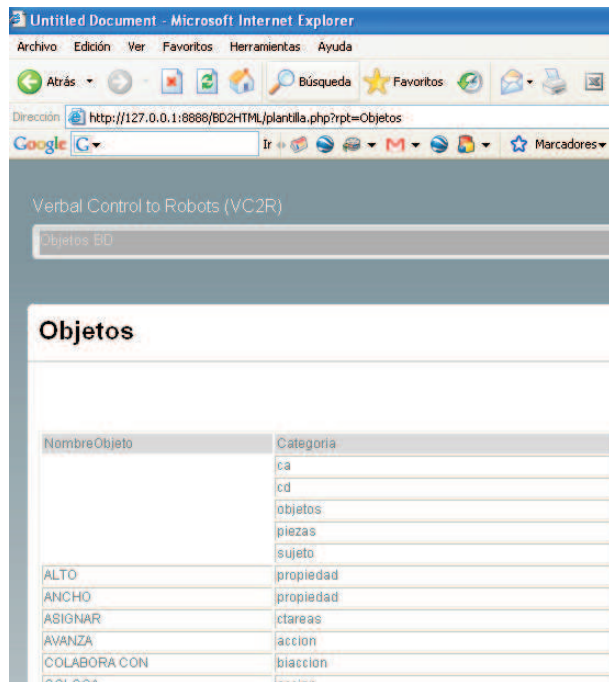


Figura 6: Interfaz de Información

que permite la multidifusión (BroadCast). De esta forma es posible tener el visor instalado en tantos ordenadores como personas se encuentren conectados. Además es posible generar una síntesis de voz de manera que se transmiten los resultados también en un lenguaje de alto nivel de una forma más efectiva.

5. CONCLUSIONES

En este artículo se ha presentado la arquitectura diseñada para el control y supervisión de un conjunto de equipos de robots que se desenvuelven en diferentes entornos. Una de las principales características de esta arquitectura consiste en la posibilidad de emplear una interfaz de alto nivel o amigable que permite que los usuarios conectados a la misma puedan emitir los comandos oportunos mediante un lenguaje de alto nivel o lenguaje natural. Asimismo, la arquitectura diseñada se encuentra estructurada en un conjunto de capas completamente distribuidas en diferentes equipos lo que posibilita una alta flexibilidad y robustez del sistema. Desde el punto de vista de empleo, esta arquitectura de teleoperación permite el acceso a múltiples usuarios de forma que puedan interactuar con múltiples entornos remotos, en cada uno de los cuales se desenvuelve un conjunto de robots.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Educación y Ciencia a través del proyecto

DPI2004-07433-C02-01 *Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots*, por el proyecto PCT-G54016977-2005 *Robots Cooperativos para la vigilancia e inspección de edificios e instalaciones industriales*, y por el proyecto GV05/012 *Planificación de tareas y entornos de aprendizaje en sistemas multi-robot cooperativos*.

Referencias

- [1] L. Cragg, P. Wo Tsui, and H. Hu. Building a fault tolerant architecture for internet robots using mobile agents. In *IORW, First British Workshop on Internet and Online Robots*, May 2003.
- [2] R.P. Ñeco, O. Reinoso, J.M. Azorín, J.M. Sabater, M.A. Vicente, and N. García. Automatic adaptation of a natural language interface to a robotic system. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2527:714–723, 2002.
- [3] R.P. Ñeco, O. Reinoso, C. Pérez, J.M. Azorín, and J.M. Sabater. A relational database design for a dynamic environment simulation tool. *Artificial Intelligence and Applications*, pages 173–177, 2003.
- [4] C. Ferrell. Failure recognition and fault tolerance of an autonomous robot. *Adaptive Behaviour*, 2(4):375–398, 1994.
- [5] K. Goldberg and R. Siegwart. Beyond web cams: An introduction to online robots. 2002.
- [6] A. Khamis, F.J. Rodríguez, and M.A. Salichs. Remote interaction with mobile robots. *Autonomous Robots*, 15:267–281, 2003.
- [7] P.X. Liu, M.Q. Meng, J. Gu, and X. Yang. A study of internet delays for robot teleoperation using biologically inspired approaches. *International Journal of Robotics and Automation*, 17(4):186–195, 2002.
- [8] R. Marin, I. Ñomdefen, R. Wirz, and P.J. Sanz. Robotics internet tele-lab: Programming using mobile devices. In *3rd International Conference on Multimedia and Information and Communication Technologies in Education*. FORMATEX, 2005.
- [9] C. Ñovales, G. Mourioux, and G. Poisson. A multi-level architecture controlling robots from autonomy to teleoperation. In *First National Workshop on Control Architectures of Robots*, Motnptellier, France, April 2006.

- [10] L. Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.
- [11] P. Saucy and F. Mondada. Khepentheweb: Open access to a mobile robot on the internet. *IEEE Robotics and Automation Magazine*, 7(1):41–47, March 2000.
- [12] D. Shulz, W. Burgard, D. Fox, S. Thrun, and A. Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics and Automation Magazine*, 7(1):48–56, March 2000.
- [13] E. Slawinski, V. Mut, and J. Postigo. Teleoperation of mobile robots. *Latin American Applied Research*, 36(2):79–86, March 2006.
- [14] P. Tsui and H. Hu. A framework for multi-robot foraging over the internet. In *IEEE International Conference on Industrial Technology*, Bangkok, Thailand, December 2002.

SCHAB: Sistema mínimo de supervisión y control de habilidades en el robot Maggie

Rafael Rivas, José María Alcérreca, Ana Corrales, Ramón Barber
RoboticsLab Universidad Carlos III de Madrid
rafael@ula.ve, josemaria@alcerreca.com, {anavalle.corrales, ramon.barber}@uc3m.es

Resumen

La incorporación de robots en nuestro entorno social es un hecho cada vez más común, muchos grupos de investigación orientan su interés en el desarrollo de robots personales con una alta interacción con usuarios sin entrenamiento previo. Para lograr tal fin los desarrollos en robótica tratan de proveer un alto nivel de autonomía. Mientras se logra la autonomía total de nuestros desarrollos es necesario diagnosticar el estado de las operaciones en el robot, el presente trabajo presenta un servicio de supervisión en uno de los robots del grupo RoboticsLab, el desarrollo se ha realizado usando componentes diseñados y creados en proyectos anteriores.

Palabras clave: Robots personales, Sistemas de Supervisión y Control, Automatización

1. Introducción

En un sistema de control, proceso industrial y en empresas de producción, es necesario que los sistemas informáticos y de control funcionen correctamente; el fallo en una mínima parte del proceso puede traer consecuencias de cualquier grado. Una manera de tener control del comportamiento de las variables y elementos que intervienen en un proceso es a través de un Sistema de Monitorización, el cual debe permitir detectar y prevenir posibles fallos en el sistema.

En el robot Maggie existen diferentes elementos y variables de proceso primordiales para su correcto funcionamiento. Posee una serie de sensores de tacto, un telémetro láser, sensores sonars, cámara de visión y servomotores, entre otros, los cuales mediante servidores en diferentes ordenadores, envían y reciben datos y señales que contribuyen en el desarrollo de las diferentes habilidades del robot.

Para garantizar la calidad de servicio se ha propuesto el desarrollo de un Sistema de supervisión para detectar cuando se ha producido la caída de un servicio, fallos en la lectura de los sensores o cambios radicales en los valores de las variables del

proceso. Con este sistema se desea mostrar el estado del robot, prevenir posibles fallos y lanzar las alarmas correspondientes cuando se produzcan.

2. El robot Maggie



Figura 1: Robot Maggie

El robot Maggie, ver figura 1, descrito en [13] es un robot personal desarrollado por el grupo de investigación RoboticsLab de la Universidad Carlos III de Madrid, su principal misión es servir de soporte a la investigación sobre interacciones Humano - Robot.

Desde el punto de vista de hardware, Maggie originalmente se ha construido a partir de un robot Magellan Pro. Hoy en día cuenta con tres ordena-

dores para el control de movimiento, visión, voz, pantalla táctil, sensores láser, ultrasonidos, infrarrojos, sensores de tacto y actuadores en la cabeza, brazos y ojos. Los ordenadores están interconectados por medio de una red Ethernet 802.3 y con el mundo exterior a través de una red WiFi 802.11.

Los desarrollos actuales de hardware están completos, por lo que espera que su aspecto exterior no varíe significativamente.

Desde el punto de vista de software, se han diseñado componentes que permiten la construcción de un sistema de control a Maggie capaz de soportar los proyectos de investigación para la cual fue diseñada y construida. Para lograr la utilización del Magellan Pro, fue necesario reemplazar el software propietario Mobility[8]. La interfaz con el hardware propietario originalmente se utilizó la biblioteca de funciones proporcionadas por Player/Stage[6] para la comunicación con el protocolo rFlex, este último es el encargado de gestionar el bus de campo que interconecta sonar, sensores infrarrojo y sensores de contacto. Sobre Player/Stage se ha desarrollado un sistema programado que proporciona acceso a las facilidades de rFlex. Hoy en día los componentes de hardware original han sido sustituidos y se han desarrollado los manejadores necesarios para utilizar los motores de desplazamiento así como el sistema de sonares, sensores infrarrojo y de contacto. Uno de los objetivos futuros es adecuar los manejadores de hardware desarrollados para ser compatibles con los desarrollos de Player/Stage.

El objetivo es construir el sistema de control del robot Maggie siguiendo el paradigma de la arquitectura A/D [2][12][11] propuesta por el grupo de investigación RoboticsLabs de la Universidad Carlos III de Madrid. En esta propuesta el componente básico de construcción es llamado habilidad [10], Las habilidades son las diferentes capacidades para razonar o llevar a cabo una acción. Estas habilidades son activadas por órdenes de ejecución producidas por otras habilidades o por un secuenciador, devolviendo datos y eventos a las habilidades o secuenciadores que las hayan activado. Esas habilidades son la base de la arquitectura AD. La figura 2 muestra un diagrama con la representación de la arquitectura A/D.

3. Planteamiento del problema

La supervisión y el control de un sistema en tiempo real es tarea fundamental para su correcto funcionamiento. Los lazos de control deben ser capaces de informar con exactitud y en un tiempo establecido y aceptable tomar decisiones adecuadas.

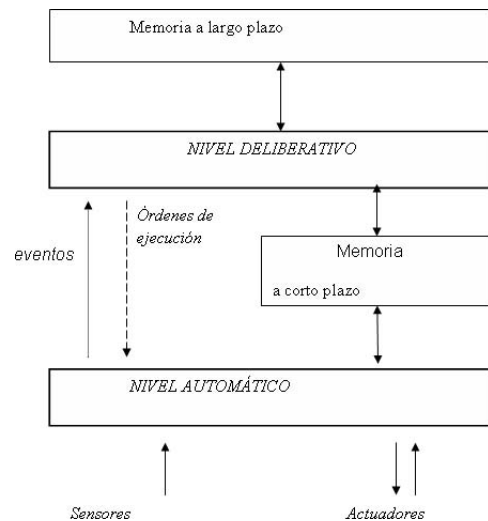


Figura 2: Capas en la arquitectura A/D

En el caso del robot Maggie hay un gran número de sensores y actuadores que en muchos casos han de funcionar en paralelo y sincronizados para un único cometido. Existen sensores cuya función es informar del entorno y sensores que informan del estado de ciertas partes del robot en un momento dado. Estos últimos serán los que cierren un lazo de control y nos den la información que trataremos en este proyecto.

En una primera aproximación uno de los problemas inmediatos es la velocidad de traslación del robot. Maggie es capaz, por ejemplo, de recorrer 0.85 metros por segundo, lo que exige estabilidad y fiabilidad en respuesta de:

- Los distintos tipos de sensores, de entorno y de velocidad.
- El procesamiento de las señales de los sensores.
- El transporte de estas piezas de información.
- El flujo de información ordenada y estructurada para garantizar su pertinencia.

Otro requisito a tener en cuenta es que el sistema sea capaz de manejar un volumen bastante grande de información y sea capaz de almacenarlo, estructurarlo y empaquetarlo de una forma optimizada para ahorrar ancho de banda. Hay que tener en cuenta que Maggie porta sensores de tal tipo capaces de enviar tantos datos desde valores representados en pocos bytes como lo es velocidad, posición, temperatura hasta imágenes completas de varios Megabytes obtenidas por medio de la cámara de vídeo.

Maggie a pesar de haber sido pensado como un robot con una alta autonomía, pero en su etapa

de desarrollo requiere que una o varias personas supervisen su funcionamiento. Por lo que se hace necesario que esta cantidad de datos sea fácilmente entendida por un humano. Para ello se necesitará una interfaz gráfica capaz de indicar de la forma más clara posible un esquema del estado del robot. Además se necesita disponer de:

- Datos exactos de cada sensor.
- Gráficas históricas.
- Estados de los controladores de cada actuador del robot.
- Estados de los servidores de datos de los sensores.
- Estado y acciones de las señales de control que emita el sistema.

Puesto que no sólo el sistema propuesto es orientado a la supervisión se han definido funcionalidades que facilitan la interacción gráfica para control del robot. Entre otros:

- Reinicialización de demonios. Tendrá control para apagar y encender cualquier controlador si se bloquean. En modo automático y en modo manual.
- Apagado de emergencia de todo o parte del sistema.
- Control manual de actuadores.

4. Alternativas en la construcción

En las primeras etapas de desarrollo se han visualizado tres etapas fácilmente diferenciables como son:

1. Recogida de los datos de los sensores
2. Transporte de los datos a un concentrador-servidor.
3. Recepción de los datos en la interfaz gráfica y exposición de los datos.

A continuación se describe cada uno de los componentes funcionales.

4.1. Adquisición de los datos de los sensores

En este caso lo que prima es la velocidad. Hay sensores como el láser que recoge un gran número de muestras por barrido, con un máximo de 720 valores cada 120 mseg.

Los drivers de los sensores están escritos en C y funcionan actualmente bajo un entorno UNIX.

Puesto que ya está implementado un modo para la depuración en los controladores de los actuadores y los sensores, ha sido necesario modificar el código del programa fuente en lenguaje de programación C. El mecanismo para enviar la información al servidor será por medio de memoria compartida, ya que es el sistema más rápido para enviar datos entre procesos en una misma máquina.

Se ha definido una estructura normalizada para este tipo de datos. El sensor escribe en memoria tantas veces como reciba datos si el rendimiento de la máquina no se ve afectado. El servidor de datos podrá leer todos o algunos de estos datos, dependiendo de la frecuencia de lectura.

Los campos de la estructura básica para un solo dato:

- Id del sensor. Entero sin signo (2 Bytes).
- Hora exacta de la muestra. Según se decida después con las pruebas de rendimiento se podrá dar más o menos exactitud a este dato. Será importante sincronizarlo después con el resto de máquinas leyendo el dato (4 Bytes).

Adicionalmente dependiendo del tipo de habilidad se agrega información adicional, en el caso de las habilidades de movimiento se anexa la velocidad actual de la base. Este mismo modelo se puede aplicar a cada uno de los procesos a supervisar. Se aprovecha la ventaja de poseer todos los programas fuentes de los diferentes servidores encargados de la interconexión con los dispositivos físicos.

4.2. Transporte de los datos a un concentrador-servidor

En la arquitectura AD se plateó la necesidad de poseer un mecanismo para compartir datos entre procesos en una ordenador o de forma distribuida entre varios ordenadores interconectados. El sistema de memoria compartido distribuida ha sido diseñado y construido, por lo que en cada ordenador de Maggie es capaz de compartir datos entre procesos activos.

El sistema distribuido de memoria compartida usa Sockets en el dominio AF-UNIX para intercambio entre procesos locales y ONC-RPC para intercambio de datos entre diferentes ordenadores. Adicionalmente se cuenta con un sistema de eventos distribuidos construidos sobre sockets AF-UNIX sobre datagramas UDP, ya que los ordenadores están sobre la misma LAN, y los datos correspondientes a cada eventos es inferior a los 8 Kbytes.

El uso de componentes de ya diseñados, construidos y probados permiten los principios de reutilización de código y por ende el ahorro en el tiempo de desarrollo.

4.3. Recepción de los datos en la interfaz gráfica y exposición de los datos

El despliegue de datos y registro de eventos especiales es tal vez la cara visible del SCHAB, para su desarrollo se poseen diferentes alternativas, alguna de ellas probadas en proyectos anteriores como [5], en este proyecto se diseñó y se construyó una interfaz para la captura de datos y manipulación de Maggie usando WebServices [1] [3]. Para la interfaz gráfica de SCHAB las alternativas son numerosas. El requisito más importante es que el lenguaje o interfaz utilizada sea capaz de obtener e intercambiar información de una manera abierta para permitir las alternativas actuales y de ser posible las futuras sin la necesidad de realizar cambios radicales en el sistema.

Alternativas:

- Macromedia Flash.
 - Ventajas:
 - Se puede lograr muy buena presentación y aspecto para el usuario, con animaciones, gráficas, etc.
 - Lenguaje ActionScript que ayudará con la presentación. Es sencillo, orientado a objetos y permite a partir de la versión 6 la interconexión por medio de sockets.
 - Se puede ver desde cualquier navegador o como aplicación independiente (standalone) sin necesidad de hacer instaladores o tener librerías para poder ejecutar otros programas.
 - El sistema de sockets además de recibir y enviar flujos de Bytes posee funcionalidades para manipular datos en XML.
 - Inconvenientes:
 - El ActionScript es un lenguaje interpretado y es de reciente creación por lo que da la impresión de no estar totalmente realizado.
- Java Applet.
 - Ventajas:
 - Es un lenguaje conocido y de amplio uso.
 - Puede desplegarse en un navegador de Internet.
 - Inconvenientes:

- La seguridad.
- Requiere conocimientos del lenguaje Java.

■ Visual Basic, Python + Gtk, Visual C++

- Ventajas
 - Son lenguajes con bastantes seguidores y gran cantidad de herramientas.
- Inconvenientes
 - las herramientas como Visual Basic o Visual C++ exigen plataformas Microsoft, todo el desarrollo de RoboticsLabs se hace sobre herramientas Linux.

Como se observa las alternativas de visualización son variadas y en muchos casos la elección apropiada dependerá de los requerimientos de la aplicación. En casos como los descritos en [5] fue necesario el uso de Applets para permitir la supervisión por medio de una página WEB, pero en la segunda versión se usó WebServices para sortear los niveles de seguridad impuestos por los filtros de tráfico (firewall) de las organizaciones involucradas en ese proyecto.

5. Prototipo realizado

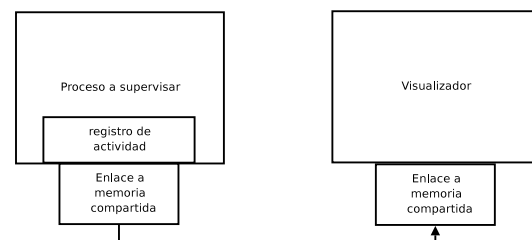


Figura 3: Funcionamiento básico del sistema SCHAB

El funcionamiento particular para la supervisión de una habilidad se muestra en la Figura 3, el funcionamiento básico consiste en colocar una instancia de la clase memoriaCompartida [9], por medio de ésta, se coloca el estado actual de la habilidad. Una vez que los datos se encuentran en la memoria compartida pueden ser adquiridos por el visualizador para su registro o envío a la interfaz del usuario. Inicialmente basta que el componente de visualización y registro compare la etiqueta con la hora exacta de la muestra y la hora actual para verificar si el sistema se encuentra activo, con un retraso en el reporte, o se presume que la habilidad se encuentra bloqueada o desactivada.

En el caso de supervisar más de una habilidad activa sobre varios ordenadores diferentes, es necesario disponer de una middleware para realizar

los servicios de distribución de los datos de las habilidades. En ese caso se activa los componentes distribuidos del sistema de memoria compartida como lo es el servidor principal y los servidores secundarios. En la figura 4 se observa la funcionalidad del mismo. La sincronización del los relojes en los ordenadores involucrados en SCHAB sincronizan sus relojes por medio de NTP [7].

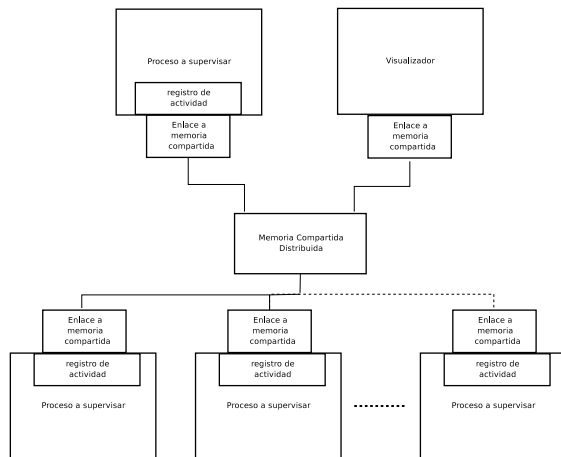


Figura 4: Funcionamiento del sistema SCHAB para la supervisión de mas de una habilidad

Por restricciones propias del sistema de memoria compartida distribuida la propagación de la información entre ordenadores conectados en el Robot Maggie implica un retardo de 300 mseg. Aunque procesos básicos de supervisión, mas no de control, este tiempo puede aceptado es importante tener en cuenta este retardo para futuras aplicaciones.

5.1. Envíos de consignas

En algunos casos es necesario activar o reinicializar algunos de los procesos que modelan las habilidades en el robot, en ese caso se ha incorporado el sistema de Eventos Distribuidos [9], este sistema implementa el en el patrón de diseño Publisher-Subscribe[4], lo que permite a un agente suscribirse a un evento específico, ante la ocurrencia de este evento de reactiva el proceso para el cual se realizo la subscripción.

6. Conclusiones

El sistema SCHAB, en su primer prototipo, ha permitido observar en línea, el estado básico de operación de los servidores de habilidades en Maggie, El desarrollo se basó en componentes creados y probados anteriormente, lo que ha permitido que la elaboración del sistema haya sido fundamentalmente la agregación de componentes.

Los diferentes niveles de visualización continúan

en pleno desarrollo y no se descarta hasta ahora ninguna de las alternativas mencionadas, el requisito fundamental es el desarrollo de interfaces que permitan el acoplamiento con el sistema de memoria compartida distribuida. Hasta ahora se ha hecho énfasis en utilizar solamente software libre sobre sistemas operativo Linux, específicamente las distribuciones Fedora C2, C3, C5 y Gentoo 2006.

Referencias

- [1] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services. Concepts, Architectures and Application*. Springer, 2004.
- [2] Ramón Barber. *Desarrollo de una arquitectura para robots móviles autónomos. Aplicación a un sistema de navegación topológica*. PhD thesis, Departamento de Ingeniería Eléctrica, Electrónica y Automática, Universidad Carlos III de Madrid, 2000.
- [3] World Wide Web Consortium. *Web Services Architecture Requirements*. <http://www.w3.org/TR/wsa-reqs/>, Visualizada por última vez en : Septiembre 2005.
- [4] Erich Gamma, Richard Helm, Raph Johnson, and John Vlissides. *Patrones de diseño. Elementos de Software*. Pearson Educación, S. A., 2003.
- [5] International Virtual Lab on Mechatronics. Home page ivlom, 2005. visited on July 2006.
- [6] Player Stage Project. Home page. <http://playerstage.sourceforge.net/>, visited on July 2006.
- [7] Network Time Protocol. Home page. visited on July 2006.
- [8] Real World Interface, Inc. *Mobility 1.1: Robot Integration Software*, 1999.
- [9] Rafael Rivas. Desarrollo de componentes básicos para el software de control de robots autónomos inteligentes. Technical report, Doctorado en Ingeniería Eléctrica, Electrónica y Automática , Universidad Carlos III de Madrid, Febrero 2006.
- [10] Rafael Rivas. Diseño y desarrollo de un sistema de control de robots autónomos inteligentes bajo la arquitectura a/d. Technical report, Doctorado en Ingeniería Eléctrica, Electrónica y Automática , Universidad Carlos III de Madrid, Febrero 2006.

- [11] Dulce Rivero. *Framework para el desarrollo de habilidades para robots móviles inteligentes basado en la arquitectura AD*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, 2004.
- [12] M. A. Salichs and R. Barber. A new human based architecture for intelligent autonomous robots. In *The Fourth IFAC Symposium on Intelligent Autonomous Vehicles*, pages 85–90, Sapporo, Japón, Sep 2001.
- [13] M. A. Salichs et al. Maggie: A robotic platform for human-robot social interaction. In *IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006)*, Bangkok, Thailand, Jun 2006.

IMPLEMENTACIÓN DE UNA HABILIDAD TTS CON EXPRESIÓN DE EMOCIONES EN UN ROBOT PERSONAL AUTÓNOMO

Elena Cal, Javi F. Gorostiza, Miguel A. Salichs
100021829@alumnos.uc3m.es, jgorosti@ing.uc3m.es, salichs@ing.uc3m.es
Universidad Carlos III de Madrid,

Resumen

En este artículo se presenta un sistema de síntesis de voz emocional. Dicho sistema se ha implementado en Maggie, el robot personal desarrollado por el RoboticsLab en la Universidad Carlos III de Madrid. Maggie ha sido diseñada para la investigación en interacción humano-robot siguiendo un nuevo paradigma de interacción que ha venido a denominarse interacción de igual-a-igual. La actual tendencia en el desarrollo de robots personales, robots sociales y robots sociables es la de dotar a estos con un conjunto de capacidades perceptivas y de actuación semejantes a las del ser humano hasta el punto de alcanzar una empatía en la interacción humano-robot semejante a la que pueda existir entre dos humanos. La importancia del trabajo que se presenta reside en el hecho de haber incorporado la capacidad de expresión hablada con carga emocional en el conjunto de capacidades de actuación de Maggie.

Palabras clave: Interacción humano-robot, interacción de igual-a-igual, emociones, Text-to-Speech.

1. INTRODUCCIÓN

Se estima que alrededor de 21500 robots de servicio de propósito singular y alrededor de 150000 robots de entretenimiento se han vendido en el año 2001 [12]. En las últimas décadas se estima que el coste medio de un robot industrial decrece entre un 5 % y un 10 % cada año [8]. El mercado en robótica en el 2003 se estima que movía unos 8 mil millones de dólares y para el año 2007 se estima que la cifra se doble [14]. El crecimiento de la producción y venta de robots de entretenimiento y robots personales no industriales va unido a un crecimiento en la investigación en la interacción humano-robot.

Numerosos robots personales cuyo propósito es la investigación en interacción humano-robot se están desarrollando en la actualidad: Kismet[7] y Leonardo [6] desarrollados en el MIT son capaces de interactuar de forma reactiva como lo haría un bebé. Sparky es un robot social que utiliza sus movimientos y expresiones faciales para in-

teractuar con humanos [16]. RUBI es otro robot antropomórfico con cabeza y brazos diseñado para investigación en interacción humano-robot en tiempo real [1] y así otros robots como Robota [4], Valerie robot recepcionista, Grace (Graduate Robot Attending a Conference) [9],... En el RoboticsLab de la Universidad Carlos III de Madrid se ha diseñado e implementado un robot personal autónomo que no solo incorpora habilidades de interacción con el entorno como la navegación, sino que también se están desarrollando habilidades de interacción social con humanos.

En la interacción humano-robot resulta esencial el modo auditivo y en especial la comunicación oral: el robot no solo debe ser capaz de reconocer y entender expresiones verbales del humano sino que también debe ser capaz de expresarse verbalmente. En la expresión verbal podemos distinguir dos aspectos: el contenido del mensaje que marca lo que el robot expresa, y su continente, que marca el cómo lo expresa. Mientras que para el primer aspecto se hace necesario la implementación de un modelo cognitivo-lingüístico en el robot que le permita construir frases con una intención comunicativa referente a un tema, para el segundo aspecto el problema está relacionado con la entonación de la voz que el robot emite. En esta entonación reside información relacionada con la relación social entre los interactores, la intención y sentido del mensaje así como el estado emocional del hablante, etc. El trabajo que aquí se expone se ha centrado en dotar al robot personal Maggie de la habilidad de expresarse verbalmente controlando la entonación o prosodia de su voz para ser capaz de expresar cinco emociones básicas: alegría, tristeza, ira, calma y confort.

En la sección 2 se hablará de un nuevo paradigma de interacción humano-robot denominado interacción de igual-a-igual. En la sección 3 se expone la arquitectura de control en la que se integra la habilidad de habla emocional que se describe en la sección 4. En las siguientes secciones se presentan resultados experimentales, conclusiones y trabajos actuales.



Figura 1: Maggie interactuando con un niño

2. PARADIGMA ROBOT-TO-HUMAN (R2H) Y HABLA EMOCIONAL

Tradicionalmente los sistemas de interacción hombre-robot se basan en una concepción maestro-esclavo, en el que el papel del operador humano es el de supervisar y dar órdenes al robot y el del robot es cumplir dichas órdenes y eventualmente suministrar al operador la información que este necesite. El robot actúa esencialmente como una herramienta manejada por el operador. En estos sistemas, la interacción con el humano actúa como un factor que limita y reduce la autonomía del robot.

El paradigma R2H pretende desarrollar robots sociales con un alto grado de autonomía, cuyo comportamiento esté condicionado por sus propios impulsos y motivaciones, que responderán en última instancia a un intento por parte del robot de maximizar o mantener en niveles de equilibrio variables internas ligadas a sus necesidades. En la arquitectura de control del robot, la interacción con los humanos se planteará exactamente con los mismos principios que la interacción con otros elementos del mundo.

El carácter social de los robots se reflejará en el hecho de que la interacción con humanos no se considerará sólo como un complemento a otras funcionalidades del robot, sino como una de sus características básicas. Esto se concretará en el desarrollo de un conjunto amplio de habilidades sociales y en que entre las motivaciones del robot se incluirán también motivaciones sociales.

El trabajo que se expone en el presente artículo corresponde a la implementación de una de estas habilidades sociales que es la habilidad de habla incorporando en la señal de voz una entonación controlada y dedicada a la expresión emocional.

Esta aproximación a la interacción entre humanos

y robots puede ser de gran interés para algunos nuevos tipos de robots, como pueden ser robots que cooperen de igual a igual con humanos, o robots dedicados a fines lúdicos, didácticos e incluso terapéuticos.

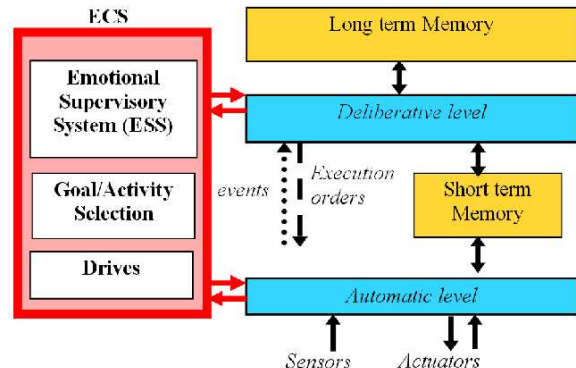


Figura 2: Arquitectura de control AD (Automática-Deliberativa)

3. ARQUITECTURA DE CONTROL AUTOMÁTICA- DELIBERATIVA

La arquitectura de control del robot personal Maggie desarrollada en [2], [3] y [5] está inspirada en los trabajos en psicología cognitiva de Shiffrin y Schneider [17] en los que se modela el razonamiento y comportamiento humano en dos niveles de inteligencia y actuación: uno automático y otro deliberativo. En la Figura 1. puede observarse el diagrama de la Arquitectura A-D, donde además se incorpora un Sistema de Control basado en Emociones desarrollado en [11]

En el nivel automático corren en paralelo varias habilidades automáticas que se caracterizan por su comunicación directa con los sensores y actuadores del robot, y por su rapidez respecto a las habilidades deliberativas. En el nivel deliberativo corre una sola habilidad deliberativa que se encarga de la planificación de la secuencia de actuaciones o de ejecución de habilidades automáticas del nivel inferior. El sistema de control basado en emociones establece el objetivo interno del robot.

La comunicación entre habilidades automáticas se realiza tanto por eventos discretos como a través de la Memoria a Corto Plazo. Una descripción detallada de la arquitectura y de su implementación en Maggie puede encontrarse en [15] y [10]

4. DESCRIPCIÓN DE LA HABILIDAD TTSskill

El sistema implementado que se presenta en este artículo tiene la forma de habilidad automática y se integra en la arquitectura de control del robot desarrollada: Arquitectura Automática-Deliberativa con Sistema de Control por Emociones (Figura 1). Básicamente la habilidad TTSskill toma como dato de entrada de la Memoria a Corto Plazo una expresión textual y teniendo en cuenta el estado emocional del robot construye una estructura prosódica o de entonación que se añade al texto para luego ser sintetizado como voz a través de los altavoces del robot. La realización de esta síntesis de voz emocional a partir de texto se realiza en los siguientes pasos: discretización del texto en fonemas, construcción de curvas de entonación y síntesis en señal de audio. La elección de estos pasos está inspirada por el trabajo realizado anteriormente en el Sony Computer Science Laboratory (CSL) [13]

Para el paso de síntesis en señal de audio, se ha utilizado una herramienta de distribución gratuita denominada MBROLA [18] Esta herramienta marca el formato en el que deben construirse los datos a lo largo del algoritmo de generación de una señal de audio a partir de texto.

4.1. Constructor fonético

Este módulo toma como dato de entrada una expresión en forma de texto o frase, y devuelve una cadena de caracteres que simbolizan la representación fonética de la frase de entrada. Dicha representación fonética corresponde con la formulación que propone MBROLA para la representación textual de fonemas hablados que puede verse en la Tabla 1.

4.1.1. Fichero fonético

MBROLA es el sintetizador TTS utilizado. En la construcción prosódica de una frase se ha de tener en cuenta su funcionamiento, es decir, el formato de los datos que esta herramienta utiliza. MBROLA recibe como entrada un fichero en formato texto estructurado por columnas que denominamos fichero fonético. En la primera columna se sitúa la serie de fonemas de la frase a sintetizar, en la segunda columna se sitúa la **duracion** de cada fonema en **ms** El resto de columnas son pares de valores. El primer valor marca un porcentaje de duración del fonema y el segundo la frecuencia en hercios a la que debe ponerse la síntesis del fonema en el porcentaje de tiempo que marca el parámetro anterior. Así por ejemplo:

Cuadro 1: Representación fonética utilizada

Fonema	Ejemplo	Representación
p	padre	paDre
b	vino	bino
t	tomo	tomo
d	donde	donde
k	casa	kasa
g	gata	gata
tS	mucho	mutSo
jj	hielo	jjelo
f	facil	faTil
T	cinco	Tinko
s	sala	sala
x	mujer	muxer
m	mismo	mismo
n	nunca	nunca
J	ajo	aJo
l	lejos	leJos
L	caballo	kabaLo
r	puro	puro
rr	torre	torre
i	pico	piko
e	pero	pero
a	valle	baLe
o	toro	toro
u	duro	duro

a 430 10 380 80 450

indica que el fonema a dura 430 ms; al 10 % de 430 ms debe alcanzar 380 Hz y debe alcanzar 450 Hz al 80 %

Puede haber tantos pares de valores porcentaje-tono como se quiera. Estos pares de valores configuran la envolvente de tono para cada fonema, por lo que cuantos más valores haya, más natural y humanizada resultará la síntesis en voz. No obstante y dado que la duración de cada fonema habitualmente es inferior al segundo, con un par de estos valores se obtienen resultados suficientemente satisfactorios. Cada fila de este fichero tiene, por tanto el siguiente formato:

fonema duracion porcentaje tono

Además de este fichero, MBROLA acepta una serie de opciones como son el volumen y el tono principal de la frase.

4.2. Constructor prosódico

Este módulo toma como dato de entrada una de los cinco estados emocionales en los que puede estar el robot: alegría, tristeza, calma, ira y confort (o neutro), construye las curvas de entonación o curvas prosódicas y adapta dichas curvas a la

longitud de la frase que va a ser sintetizada como voz emocional. Para caracterizar la prosodia de cada frase y teniendo en cuenta el formato del fichero fonético que acepta MBROLA, podemos enumerar las siguientes variables físicas: por un lado y en relación al conjunto entero de fonemas de la frase: el tono principal, envolvente de tono y volumen principal; y por otro lado y asociado a cada fonema de la frase: una envolvente de tono dentro de cada fonema y una duración de fonema. El problema es dar valores a estas variables para que la prosodia de la frase sintetizada exprese una emoción concreta y no otra. Para alcanzar este objetivo se define un conjunto de parámetros intuitivos que sirven como marco entre la emoción que se quiere expresar y las variables físicas enumeradas arriba. En la tabla x. se muestran los valores escogidos para este conjunto de parámetros prosódicos en función de la emoción que se quiere expresar mediante voz. Estos parámetros se describen a continuación:

- **VOLUMEN**(sin unidades) establece volumen sonoro de la totalidad de la frase sintetizada. Es el factor por el que se multiplica el volumen que por defecto utiliza MBROLA en la construcción del fichero de audio correspondiente a la frase de voz sintetizada.
- **MEANDUR**(ms) es la duración principal de cada fonema.
- **DURVAR** (ms) marca el rango de variación de la duración de cada fonema.
- **MEANPITCH**(Hz) es la frecuencia principal de cada fonema.
- **PITCHVAR** (Hz) marca el rango de variación de la frecuencia principal de cada fonema.
- **LAST-ACCENT** (0 — 1) parámetro booleano que marca si el último fonema de la frase se va o no a acentuar.
- **CONTOUR** (RISING — FALLING) parámetro booleano que marca el contorno por defecto del tono de cada fonema.
- **LAST-CONTOUR** (RISING — FALLING) parámetro booleano que marca el contorno del último fonema.

4.2.1. Algoritmo del constructor prosódico

El fichero fonético, entrada para MBROLA, es construido en formato texto a partir de la lista de fonemas que devuelve el constructor fonético y de la lista de parámetros que devuelve el constructor

prosódico. Así, el fichero es completado con la lista de fonemas de la frase concreta que se va a sintetizar y con los valores numéricos correspondientes y explicados anteriormente para marcar la prosodia de la frase: **duración**, **porcentaje** y **tono**. A continuación se describe el algoritmo que construye la duración y tono de cada fonema. Este algoritmo va recorriendo todos y cada uno de los fonemas.

```

duracion( fonema ) = MEANDUR + rand( DURVAR )
if( fonema es consonante )
    tono( fonema ) -= PITCHVAR
if( fonema es vocal )
    tono( fonema ) += PITCHVAR
    if( fonema == accentuado):
        duracion( fonema ) += DURVAR
if( CONTOUR == RISING)
    if( fonema es consonante )
        tono( fonema ) += PITCHVAR
    if( fonema es consonante )
        tono( fonema ) -= PITCHVAR
if( CONTOUR == FALLING)
    if( fonema es consonante )
        tono( fonema ) -= PITCHVAR
    if( fonema es consonante )
        tono( fonema ) += PITCHVAR
if( LAST-ACCENT ) para última palabra
    e = PITCHVAR / 2
    duracion( fonema ) += DURVAR
    if( LAST-CONTOUR == FALLING)
        pitch( fonema ) -= ( fonema +1 ) * e
    if( LAST-CONTOUR == FALLING)
        pitch( fonema ) -= ( fonema +1 ) * e
    e = 2 * e
porcentaje(fonema) = 80 + rand( 20 )

```

Cabe destacar el uso de la función de aleatoriedad con el propósito de introducir en la síntesis cierto carácter natural característico del habla, es el caso, por ejemplo del valor del parámetro **porcentaje**

4.3. Sintetizador TTS

MBROLA es un proyecto iniciado en el Laboratorio de Teoría de Circuitos y Tratamiento de la Señal (TCTS Lab) de la Facultad Politécnica de Mons (Bélgica) cuyo propósito principal es el de implementar sintetizadores TTS para todas las lenguas posibles. Es una herramienta gratuita pero cuyo código no está abierto, sin embargo, su adaptación a una aplicación externa es sencilla. MBROLA consta de un ejecutable y una base de datos en formato binario asociada a una lengua y a un agente o personaje que marca tanto el idioma (y por tanto el conjunto de fonemas posibles)

Cuadro 2: Parámetros prosódicos para cada emoción

Parámetro	alegría	tristeza	calma	ira	confort
VOLUMEN (s.u.)	0.8	1	1	2	2
MEANDUR (ms)	170	300	200	150	300
DURVAR (ms)	50	100	100	20	300
MEANPITCH (Hz)	600	250	370	100	350
PITCHVAR (Hz)	100	30	10	50	50
LAST-ACCENT (0 1)	1	0	0	0	1
CONTOUR	RISING	FALLING	RISING	FALLING	RISING
LAST-CONTOUR	RISING	FALLING	RISING	FALLING	RISING

como el tono o soniquete del habla. Hay personajes femeninos y masculinos. El ejecutable que trae MBROLA recibe como opciones el volumen y el tono general de la expresión hablada y como entrada el fichero asociado a la base de datos del idioma y el que hemos venido a denominar fichero fonético con la lista de fonemas a sintetizar y parámetros acústicos antes presentados. Este ejecutable devuelve un fichero de audio. Los formatos permitidos son .wav o .au

El servidor TTSServer que corre dentro de la habilidad TTSSkill se encarga de ir leyendo de la memoria compartida una frase nueva que haya de ser sintetizada; toma la emoción con la que debe expresarse y construye el fichero fonético; éste se compila con el ejecutable de MBROLA y la base de datos asociada al idioma de lo cual se obtiene un fichero de audio que se lleva al dispositivo de audio para que se oiga.

5. RESULTADOS EXPERIMENTALES

La conexión de la habilidad implementada TTSSkill con el resto de habilidades y demás entidades de la arquitectura (ver Figura 1) es totalmente satisfactoria. Esto se debe a que la estructura software para la implementación de una habilidad automática en la arquitectura AD se ha estandarizado en previos trabajos. El funcionamiento en tiempo real es también satisfactorio: TTSSkill es capaz de realizar la transformación de la frase textual a una lista de fonemas, el cálculo de los parámetros prosódicos y la síntesis de la onda de audio generada con suficiente rapidez como para no apreciarse un retardo digno de mayor estudio.

La habilidad automática TTSSkill se encarga de la síntesis de texto a voz con entonación emocional, por tanto, el resultado final de esta implementación es un resultado audible. En este apartado, se va a presentar una breve comparación de las características sonoras de distintas frases al haber sido procesadas de modo distinto para expresar

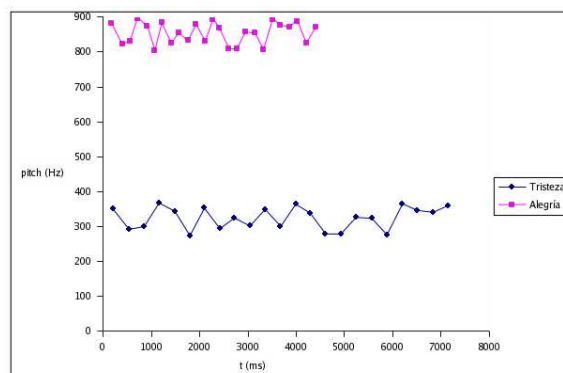


Figura 3: Comparación del tono de cada fonema a lo largo de las frases: "Soy Magui y estoy muy triste"(abajo) y "Soy Magui y estoy muy contenta"(arriba)

emociones distintas.

En la Figura 4 puede observarse la representación gráfica de la variación del tono a lo largo de cada fonema de la frase en función del tiempo para dos emociones antagónicas: alegría y tristeza. Para el caso en el que se quiere expresar alegría el texto de la frase a sintetizar fue "Soy Maggie y estoy muy contenta" En el caso de la expresión de tristeza fue "Soy Maggie y estoy muy triste". Por tanto, teniendo en cuenta la tabla 1, la lista fonética del primer caso quedaría `s o i m a g i i e s t o i m u i k o n t e n t a` (24 fonemas) `s o i m a g i i e s t o i m u i t r i s t e` (22 fonemas) En el caso de la expresión de alegría puede observarse que el tono principal es mayor que en la expresión de tristeza. Además la variación del tono entre fonemas es más rápida, dado que la duración de cada fonema es menor.

6. CONCLUSIONES Y DESARROLLOS ACTUALES

Se ha diseñado e implementado una habilidad automática TTSSkill para un robot personal autónomo incorporando así en el robot la capaci-

dad de transformar texto a habla. Esta habilidad está encapsulada en la arquitectura automática-deliberativa y por tanto es susceptible de ser utilizada por el resto de habilidades de la arquitectura que funcionan concurrentemente en el robot. A la voz convertida desde texto se le añade, además, un carácter emocional, lo cual es de gran importancia dado que el robot personal está diseñado para la interacción humano-robot en el nuevo paradigma "interacción de igual-a-igual" El robot es así capaz de expresar sus emociones con el tono de la voz.

El sistema implementado permite añadir al robot con relativa facilidad otras capacidades vocales como es la capacidad de entonar para realizar preguntas o exclamaciones. Así mismo actualmente se está diseñando un sistema que incorpore en el robot la habilidad de cantar utilizando como base el sistema presentado en este artículo.

Agradecimientos

Los autores quieren agradecer sinceramente los fondos cedidos por el Ministerio de Educación y Ciencia (MEC) para el proyecto denominado "Interacción Humano-Robot de Igual-a-Igual" (R2H), del MEC (Ministerio de Educación y Ciencia)

Referencias

- [1] J. Chenu B. Fortenberry and J. Movellan. Rubi: A robotic platform for real-time social interaction. In *Third International Conference on Development and Learning (ICDL'04)*, 2004.
- [2] R. Barber. *Desarrollo de una arquitectura para robots móviles autónomos. Aplicación a un sistema de navegación topológica*. PhD thesis, Universidad Carlos III de Madrid, Spain, 2001.
- [3] Ramón Barber and Miguel Ángel Salichs. A new human based architecture for intelligent autonomous robots. In *The Fourth IFAC Symposium on Intelligent Autonomous Vehicles*, pages 85–89, 2001.
- [4] A. Billard, K. Dautenhahn, and G. Hayes. Robota: Clever toy and educational tool. *Robotics & Autonomous Systems*, (42):259–269, 2003.
- [5] M. J. López Boada. *Sistema de Control para Robots Móviles Autónomos Basado en Habilidades Reactivas*. PhD thesis, Universidad Carlos III de Madrid, Spain, 2002.
- [6] C. Breazeal, A. Brooks, D. Chilongo, J. Gray, A. Hoffman, C. Kidd H. Lee, J. Lieberman, and A. Lockered. Working collaboratively with humanoid robots. *ACM Computers in Entertainment*, 2(3), July 2004.
- [7] C. Breazeal and B. Scasselatti. Infant-like social interactions between a robot and a human caretaker. Technical report, MIT, Artificial Intelligence Laboratory, 1998.
- [8] Brian Carlisle. Robotics: Doing well, will do better. *Manufacturing Engineering*, 130(5), May 2003.
- [9] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B.P. Sellner, R. Simmons, K. Snipes, A. Schultz, and J. Wang. Grace and george: Social robots. In *AAAI. Proceedings of AAAI'04. Mobile Robot Competition Workshop*, number Technical Report WS-04-11, pages 15–20, August 2004.
- [10] Javi F. Gorostiza et al. Multimodal human-robot interaction framework for a personal robot. In *RO-MAN 06: The 15th IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, U.K., September 2006. IEEE.
- [11] María Malfaz and Miguel Ángel Salichs. A new architecture for autonomous robots based on emotions. Lisbon, Portugal, July 2004. Fifth IFAC Symposium on Intelligent Autonomous Vehicles.
- [12] International Federation of Robotics (IFR) and The United Nations Economic Commission for Europe. World robotics 2002 - statistics, market analysis, forecast, case studies and profitability of robot investment. Geneva, Switzerland: United Nations, 2002.
- [13] Oudeyer Pierre-Yves. The production and recognition of emotions in speech: Features and algorithms. *International Journal of Human Computer Interaction: special issue on Affective Computing*, (59(1-2)):157–183, 2003.
- [14] William Pritchard. *Robots Automation Devices*.
- [15] Miguel Ángel Salichs et al. Maggie: A robotic platform for human-robot social interaction. In *IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006)*, Bangkok, Thailand, June 2006. IEEE.
- [16] M. Scheeff. Experience with sparky: A social robot. In *Proceedings of the Workshop on Interactive Robot Entertainment*, 2000.

- [17] R. Shiffrin and W. Schneider. Controlled and automatic human information processing: Ii perceptual learning, automatic attending and a general theory. *Psicological Review*, (84):127–190, 1997.
- [18] MBROLA Project Development Team. Mbrola project, 1996.

SISTEMA DE VISIÓN PARA EL AISLADO EN VACÍO DE OBRAS DE ARTE Y OTROS DOCUMENTOS BASADOS EN PAPEL

Joan Vidal Llidó, UPV, Vidal.Joan@gmail.com

Antonio-José Sánchez-Salmerón, ai2, UPV, asanchez@isa.upv.es

Carlos Ricolfe Viala, ai2, cricolfe@isa.upv.es

Martin Mellado Arreche, ai2, UPV, martin@ai2.upv.es

Salvador Muñoz Viñas, crbc, UPV, smunoz@crbc.upv.es

Resumen

En el proceso de restauración de obras de arte basadas en papel requiere de un proceso de hidratación de las fibras del papel para dotarles de flexibilidad. Posteriormente en la fase de secado se realiza un seguimiento visual del proceso de contracción para poder controlar la forma final del lienzo. Hasta este momento, todas estas pruebas de seguimiento visual se realizaban por los técnicos especialistas. Los resultados obtenidos mediante este procedimiento humano están sujetos a variaciones las cuales son ajenas totalmente a los procesos. Estas variaciones están provocadas por el criterio subjetivo que cada técnico aporta a la medida, las cuales se ven afectadas a su vez por el estado de ánimo en que se encuentra el mismo. Para obtener unos resultados más homogéneos y precisos es necesario desarrollar unos algoritmos de seguimiento visual automática los cuales estandaricen el proceso de medida necesario. El presente artículo presenta varias técnicas de seguimiento visual semi-automatizadas del proceso de secado en restauración de obras de arte basadas en papel.

Palabras Clave: Seguimiento visual, control de procesos, restauración de obras de arte.

1 OBJETIVOS

Los tratamientos de restauración de obras de arte basadas en papel son procesos complicados que se desarrollan en varias etapas. El alisado es una de los procedimientos más delicados. Generalmente la hoja requiere una hidratación previa para que las fibras que lo componen adquieran la flexibilidad necesaria para permitir su manipulación. Después se procede a su inmovilización y secado.

Sin embargo, detrás de este principio tan simple se esconden problemas técnicos de consideración. Cuando las moléculas de agua forman puentes de hidrógeno con las moléculas de celulosa, el tamaño de la hoja aumenta. Del mismo modo, cuando se produce el secado, el tamaño de la hoja disminuye. Hay muchos factores que determinan el tamaño final

de la hoja: grado de inmovilización, tipo de fibra, tensiones latentes, tipo de entintado, etc. En resumen, las fibras que forman el papel reaccionan de una manera no estimable completamente, hecho que imposibilita el cálculo exacto de las dimensiones de la obra una vez alisada. La deformación experimentada por la hoja puede (de hecho, suele) no ser uniforme a lo largo de su estructura y puede incluso deformarse en una dirección y no hacerlo en absoluto en la dirección perpendicular.

En el Instituto de Restauración del Patrimonio de la Universidad Politécnica de Valencia (UPV) se está investigando este fenómeno, con el fin de comprender este fenómeno y, en último término, desarrollar un sistema que permita paliar los problemas técnicos que de él se derivan. Para observar y monitorizar las alteraciones dimensionales de las hojas se solicitó la ayuda del grupo de investigación de robótica del Instituto de Automática e Informática Industrial de la UPV. Los requerimientos eran complejos: se trataba de desarrollar un sistema de visión artificial sencillo, flexible y de bajo coste, para cuantificar y registrar en tiempo real las alteraciones en las dimensiones de una hoja alisada bajo presión. El sistema debía incorporar también un sistema de detección de umbrales, para permitir la aplicación sistemáticas de técnicas correctivas de alisado.

Lamentablemente, esto no es sencillo. Los sistemas de inmovilización de la hoja basados en presión (prensas de mano, prensas hidráulicas, etc.) emplean materiales opacos, que hacen muy difícil registrar los movimientos.

Sin embargo, en la UPV también se están desarrollando sistemas basados en la aplicación de vacío que permiten un cierto grado de visualización de la hoja. Con todo, en estos sistemas la hoja está separada del observador por una lámina de plástico siliconado y otra de tejido-no-tejido de monofilamento de nylon. En condiciones de trabajo, la primera lámina genera brillos, y no se mantiene lisa; la segunda hoja, por su parte, permite sólo una visión muy borrosa de la hoja (véase Figura 1).

En resumen, el objetivo de esta investigación es construir un sistema de visión artificial basado en cámaras web y en un PC que realizará las etapas de análisis de imagen en las condiciones descritas arriba (dos capas de interposición entre el objeto y la cámara, bajo coste, flexibilidad). El sistema debería ser capaz de medir las deformaciones en las hojas de colores variables y advertir cuando estas variaciones empiezan o superan un cierto umbral. Además, debería ser eficaz en condiciones de iluminación poco controladas.

Para el seguimiento de las deformaciones que se producen en las hojas, se han implementado varias técnicas de seguimiento de características. Podemos dividir las técnicas a implementar en dos: seguimiento directo y seguimiento por correspondencia de bordes [5]. Los desplazamientos medidos se guardarán en un archivo de texto tablado importable por Excel sin ninguna complicación.

2 EL SISTEMA DE VISIÓN

2.1 EL SISTEMA DE SECADO

El sistema de secado se basa en una superficie rígida sobre la que se coloca material de absorción, la hoja a secar, una lámina de tejido-no-tejido de monofilamento de nylon y una lámina de plástico siliconado, transparente y no poroso. El vacío se genera entre la mesa y la lámina de plástico no poroso, de modo que la hoja a alisar es presionada contra la superficie rígida.

El observador puede ver la hoja a través de ambas láminas de plástico, pero las condiciones distan de ser óptimas (Figura 1). Como además las dimensiones de cada hoja son variables, es necesario que la cámara se puede mover con comodidad por toda la superficie de alisado.



Figura 1: La hoja durante el proceso de secado

2.2 EL SISTEMA DE CÁMARAS

El sistema de visión está compuesto por un

determinado número de cámaras web conectadas a través de una interfaz USB con un PC de proceso. Estas cámaras se sitúan sobre la plataforma de secado con la ayuda de minitripodes y controlan esquinas o puntos importantes de la hoja en busca de desplazamientos.

La figura 1 está tomada directamente de una captura de imagen de un proceso de restauración determinado en la que se ve la esquina de la hoja a secar. (Se pueden ver dos hojas en proceso de secado, una a la derecha y otra a la izquierda).

3 SOLUCIONES PROPUESTAS

Las soluciones propuestas están basadas en la aplicación de diferentes técnicas de análisis de imágenes [2], [3].

3.1 CORRESPONDENCIA DIRECTA

La primera de las soluciones propuestas es la que nombramos como correspondencia directa y consiste en capturar una fracción cuadrada del primer fotograma devuelto por cada cámara (fracción que será escogida por el usuario) y entonces se busca una fracción de medida equivalente con una similitud máxima pixel a pixel en las siguientes imágenes capturadas. Este proceso se puede realizar sobre las imágenes originales o sobre las mismas aplicando un filtro Sobel. El método es bastante robusto, aunque es sensible a variaciones de luz, brillos, etc.

3.1.1 Seguimiento

En este método se captura una fracción cuadrada de dimensión N de la imagen original I_0 , con un desplazamiento (l, m) elegido por el usuario. A esta imagen la nombramos F y $F_{(x,y)}$ a los valores de sus pixels, donde $(x, y) \in [0, N-1]$. La formación de F se muestra a continuación en la expresión (1):

$$F_{(x,y)} = I_{0(x+l,y+m)} \forall ((x,y) \in [0, N]) \quad (1)$$

Posteriormente, buscamos para cada imagen posterior I_L ($L \in [1, Z]$) un par de valores (l_L, m_L) que minimicen la siguiente expresión (2):

$$G_L = \sum_{i < N} \sum_{j < N} \left(\left| F_{(i,j)} - I_{L(i+l_L, j+m_L)} \right| \right) \quad (2)$$

La diferencia registrada entre (l, m) y (l_L, m_L) determina el desplazamiento del punto de interés seleccionado por el usuario en la imagen I_L , de forma que su distancia euclídea representa el módulo de su movimiento en pixels.

3.1.2 Actualización de la fracción a seguir

Puede darse el caso que debido a la variación de las condiciones ambientales en la captura de una imagen I_L y siguientes, el valor mínimo registrado para (2) sea muy elevado, en estos casos es necesario volver a calcular F de la forma que se muestra a continuación en la expresión (3):

$$F_{(x,y)} = I_{L(x+l_L, y+m_L)} \forall ((x, y) \in [0, N]) \quad (3)$$

El problema que se plantea en este punto es el cálculo de un umbral a partir del cual se decida actualizar la fracción de imagen F .

Con una cámara perfecta y unas condiciones ideales bastaría con elegir el par (l_L, m_L) de forma que el mínimo G_L según la expresión (1) fuera distinto de 0, pero en la realidad, y debido principalmente al ruido blanco presente en todo sensor, necesitamos un enfoque algo más complejo.

Podemos determinar la distribución del ruido blanco presente en las imágenes tomando una imagen oscura sin iluminación con la cámara funcionando en las mismas condiciones que lo hará posteriormente y entonces, toda señal presente será ruido.



Figura 2: Histograma de una imagen oscura

En la figura 2 podemos ver el histograma de frecuencias de una imagen oscura y podemos distinguir que su forma es la característica de una Gaussiana, en este caso con media 5,9 y una desviación típica de 4,8.

Ahora debemos escoger un umbral que no provoque una sobreactualización de F , ya que, dado que las variaciones dimensionales de las hojas pueden tomar valores del orden de subpíxel, puede ser una fuente de errores de medida. Dado que el ruido presente tiene una distribución Gaussiana el 99% de las muestras se encuentran presentes dentro de 2,58 veces la desviación típica a los lados de la media [4]. Escogiendo un umbral para el G_L mínimo de $(4,8 * 2,58) * n$, siendo $n = N^2$, nos aseguramos de evitar la sobreactualización en el 99% de los casos.

3.2 SEGUIMIENTO DE FRONTERAS

Otra solución probada ha sido la extracción de las fronteras que aparecen en el primer fotograma capturado mediante un filtro de Canny [1], fronteras que serán comparadas con las extraídas de los fotogramas posteriores con un algoritmo de correspondencia de cadenas. Una vez comparadas las cadenas, podemos establecer que fronteras de una imagen equivalen a las de otra y medir así sus desplazamientos. Este método presenta una mayor tolerancia a las variaciones de luz y condiciones externas que el método anterior, aunque el coste computacional es mucho más elevado.

Este método se ha enfocado de dos formas, en la primera se selecciona una frontera de la primera imagen y se buscaban “equivalentes” en las imágenes siguientes. La frontera escogida debe ser una frontera fuerte y larga, que a la vez tenga poco parecido con las otras fronteras que puedan aparecer. En el segundo enfoque se seleccionan un grupo de fronteras de la primera imagen y se buscan en las imágenes siguientes fronteras que superen cierto umbral de parecido para cada una de las primeras.

3.2.1 Una frontera

Dado que la frontera que elegimos para seguir debe tener las propiedades de ser fuerte (F), larga (L) y poco parecida (P) a las demás; nos encontramos ante una función multiobjetivo M , en la que debemos ponderar la contribución de cada uno de sus factores al resultado final (4).

$$M = A * F + B * L + C * P \quad (4)$$

Estos factores se han calculado experimentalmente usando prueba y error, llegando a la conclusión que el grado de similitud devuelto por el algoritmo de correspondencia de cadenas depende ligeramente de las características de estas dificultando la obtención de un valor apropiado para C .

3.2.2 Varias fronteras

A diferencia del método anterior, aquí no es necesario la elección de fronteras que tengan poca similitud entre ellas, nos limitamos a usar las fronteras más fuertes y largas, por lo que la función multiobjetivo anterior queda sin el último sumando para este método.

Al comparar un conjunto de fronteras inicial F_0 con otro conjunto F_L , puede darse el caso de aparición de multicorrespondencias entre varias fronteras similares, esto no representa un problema ya que posteriormente se representa el desplazamiento

registrado para cada correspondencia encontrada sobre un plano, generando una nube de puntos que representa el desplazamiento para cada frontera como se puede ver en la figura 3:

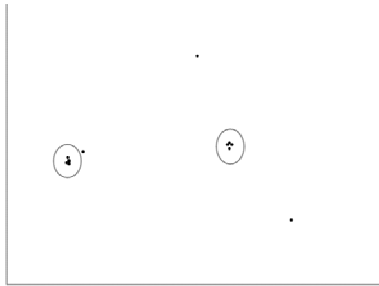


Figura 3: Nubes de puntos representando desplazamientos de fronteras

En la figura anterior se pueden observar unas nubes de puntos representando el desplazamiento de dos objetos (las dos circunferencias) y puntos aislados originados por correspondencias erróneas entre fronteras.

El último problema que plantea este método es la necesidad de encontrar las agrupaciones de puntos que representan verdaderos desplazamientos de objetos, para ello hemos escogido un umbral de distancia (H) y ponderado cada punto del conjunto (J) según la expresión (5):

$$P(J_h) = \sum_{i=0}^N V(J_i) : d(J_h, J_i) < H \quad (5)$$

Siendo d la distancia euclídea y V una función que devuelve la magnitud de la frontera que originó el punto J_i . Posteriormente se debe recorrer el conjunto de puntos J aceptando como válidos los desplazamientos que cumplan con la siguiente expresión (6):

$$J_h : \forall J_i \in J, (V(J_i) < V(J_h)) \vee (d(J_i, J_h) > H) \quad (6)$$

Finalmente, a cada selección de un punto J_s , se deben eliminar del espacio todo desplazamiento que cumpla con (expresión (7)):

$$J_h : \forall J_i \in J, d(J_h, J_i) < H \quad (7)$$

Todos los puntos seleccionados serán considerados como desplazamientos de un objeto en la imagen.

3.3 UMBRALIZACIÓN

Esta técnica consiste en umbralizar la imagen para

poder separar la hoja del fondo y una vez hecho esto calcular los bordes de la hoja recorriendo las fronteras entre regiones creadas. Esta técnica, aunque bastante simple, no se ha llegado a implementar ya que la baja diferencia de color entre el fondo y la hoja causaba una aparición frecuente de errores.

4 PROBLEMAS ENCONTRADOS

4.1 LA CAPA SEMITRANSARENTE

La lámina de monofilamento de nylon ha sido el elemento que más problemas ha provocado, ya que su transparencia no es uniforme a lo largo de su estructura, sino que presenta unas fibras más opacas superpuestas a una superficie más transparente. La presencia de fibras introduce unas elevaciones en brillo del papel que no son completamente estimables y por tanto resultan difíciles de eliminar.

Para el tratamiento de este problema se han implementado varias técnicas, desde la aplicación de filtros FIR paso banda con ventanas de Boxcar y Hamming hasta el análisis en frecuencia de las imágenes, pero el mejor resultado se ha obtenido aplicando un filtro de mínimos con un kernel de 5x5 seguido de un emborronado Gausiano. La eliminación total de las fibras superpuestas no ha sido posible, aunque si que se han podido suprimir en gran medida.

4.2 LA ETAPA DE CÁLCULO

En la mayoría de aplicaciones sólo existe un hilo de proceso, por tanto, la interfaz gráfica queda bloqueada temporalmente cuando es necesaria una etapa larga de proceso por parte de la CPU. El bloqueo temporal de una aplicación siempre genera cierto malestar en el usuario, ya que el programa tarda cierto tiempo a responder a las “exigencias” de su operador. Este problema se agrava cuando la etapa de cálculo se alarga dejando la aplicación “bloqueada” durante un largo periodo de tiempo.

El problema anterior se ha resuelto fácilmente con la creación de un hilo independiente encargado de las tareas de proceso; de esta forma el hilo de proceso principal se encarga de atender al usuario y delega todo el trabajo pesado en un segundo hilo, que puede ser “matado” o detenido a conveniencia.

5 RESULTADOS OBTENIDOS

5.1 CORRESPONDENCIA DIRECTA

Esta técnica de correspondencia directa, ya sea sin procesar las imágenes o aplicándoles filtros, es un

método bastante simple, aunque se ha demostrado eficaz para realizar el seguimiento de los bordes de las hojas. Por su forma de trabajar resulta sensible a las variaciones de iluminación. La aparición de brillos también puede ser un problema, pero sólo cuando estos se superponen a la zona que el algoritmo se encuentra siguiendo. Con esta técnica hemos conseguido unos buenos resultados pese a su relativa sencillez.

5.2 SEGUIMIENTO DE BORDES

Este método ha sido, de largo, el más complejo de los implementados. Esta técnica es prácticamente inmune a las condiciones de iluminación, aunque es sensible a la aparición de brillos sobre cualquier parte de la imagen. Se trata de la técnica más robusta ya que es capaz de detectar cualquier tipo de movimiento producido, consumiendo, eso sí, mucho más tiempo de CPU que cualquier los otros métodos propuestos.

En la práctica se ha mostrado poco viable para este caso, ya que la capa de monofilamento de nylon introduce bordes ficticios que dificultan en gran medida la detección de bordes por parte del detector usado (Canny). Aunque sí que es capaz de realizar su tarea cuando no existen interferencias en la imagen.

5.3 UMBRALIZACIÓN DE LA HOJA

Esta técnica se descartó antes de ser implementada, debido a la dificultad de encontrar un umbral que permitiera una separación de la hoja respecto del fondo en cualquier caso, ya que el color del papel no es conocido en tiempo de diseño de la aplicación y podría darse el caso que tuviera gran similitud con el fondo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos de la UPV PPI de referencia 20040897 y 20040908, por el proyecto europeo MASMICRO, Project number 500095-2, por los proyectos FEDER-CICYT con referencia DPI2005-09133-C03-01 y DPI2005-08732-C02-02 del Ministerio de Educación y Ciencia, así como por el proyecto de investigación de la Generalitat Valenciana, Consellería de Empresa, Universidad y Ciencia GV06/115.

Referencias

- [1] Canny J.F. *A computational approach to edge detection*. IEEE Trans. on Pattern Analysis and Machine Intelligence, 8 (6), pp 679-698, 1986.
- [2] Davies, R. (1997). *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press.

- [3] Gonzalez R.C. and Woods, R.E. (1992). *Digital Image Processing*. Addison-Wesley reading, MA, 1992.

- [4] <http://www.digitalreview.com.ar/distribucion-normal/>

- [5] Vidal Lidó, Joan. *Sistema de visió per al control de l'aïllat en buit d'obres d'art i altres documents basats en paper*. Proyecto Fin de Carrera. F.I. Universidad Politécnica de Valencia, 2006.

BÚSQUEDA SELECTIVA DE OBJETOS EN ENTORNOS NO ESTRUCTURADOS MEDIANTE APARIENCIA MULTICANAL Y MULTIESCALA

M.Asunción Vicente, César Fernández, Arturo Gil, Ramón Neco, Luis Payá, Oscar Reinoso
 Universidad Miguel Hernández, Dpto. de Ingeniería de Sistemas Industriales
 Avda. de la Universidad s/n 03202 Elche (Alicante), España.
 suni@umh.es

Resumen

Se presenta un sistema de detección de objetos basado en apariencia multicanal y multiescala, cuya característica principal es su capacidad para trabajar en entornos no estructurados

Palabras clave: visión artificial, detección de objetos, apariencia, PCA, NMFSC.

1 INTRODUCCIÓN

El objetivo final de la aplicación que presentamos es la detección de un objeto cualquiera en un entorno no estructurado. Básicamente, la tarea de la *detección de objetos* consiste en localizar regiones en la imagen de test que muestren al objeto de interés. Tradicionalmente, los detectores más populares están orientados a la detección de objetos genéricos como caras [5], [7] y [10] o automóviles y sus matrículas [4]. Aunque también existen algunos trabajos que estudian el problema intentando obtener métodos válidos para objetos específicos, como en [3] y [6].

Nuestra propuesta consiste en un detector de objetos específicos de forma libre basado en **apariencia** utilizando **características multicanales**, adaptable al objeto, es decir, un algoritmo que permita su uso con objetos distintos. La figura 1 ilustra el objetivo que persigue nuestra propuesta.

El algoritmo de detección de objetos basado en apariencia más eficaz, pero de alto coste computacional, consiste en la división de la imagen original en múltiples ventanas de diferentes tamaños (diferentes escalados) para posteriormente, realizar la correlación de cada una de estas ventanas con la imagen o imágenes representativas del objeto [1]. La ventana que presente mayor correlación con la imagen modelo del objeto es la señalada como la que contiene al objeto de interés. Es evidente que este método tan simple funciona eficazmente si el objeto se encuentra aislado del resto de objetos de la escena y el fondo es uniforme y similar al utilizado en las imágenes de entrenamiento (su modelo de apariencia), sin embargo, en escenarios com-

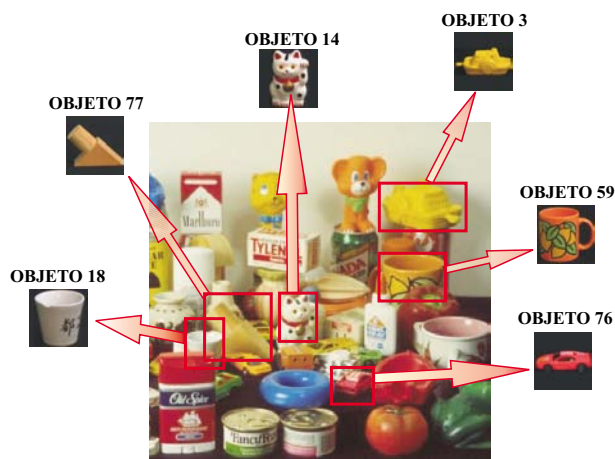


Figura 1: Tarea a realizar por el detector de objetos: localizar regiones en una escena que muestren al objeto u objetos de interés. En esta figura se muestra una escena compleja con múltiples objetos de la librería COIL-100.

plejos, su funcionamiento puede no ser del todo correcto debido a la presencia de fondos diferentes al de las imágenes de entrenamiento, oclusiones, agrupamientos con otros objetos y diferencias en la iluminación. Otra desventaja considerable es su alto coste computacional: aún estando el objeto aislado en la escena, para su correcta detección se deben obtener ventanas del tamaño adecuado para que el objeto se encuentre presente de forma similar a las imágenes de entrenamiento, por lo que el procedimiento de obtención de ventanas de distintos tamaños hasta conseguir aquella que presente una correlación mayor puede ser largo y tedioso si se utilizan las imágenes de entrenamiento en el espacio imagen, es decir, si no se utiliza ningún tipo de compresión de las ventanas y de las imágenes modelo del objeto.

Nuestra propuesta se fundamenta en este método de detección basado en apariencia con ventanas; sin embargo, es posible hacer uso de menos información en las imágenes mediante la compresión por extracción de características [8], lo que reduce drásticamente el coste computacional de los clasificadores utilizados para realizar la correlación,

u otra medida de distancia, entre las diferentes imágenes. Las principales novedades del método propuesto son las siguientes:

- Combinación de *dos etapas de búsqueda en paralelo*, para detectar con más robustez y ahorrar coste computacional.
- Utilización de dos métodos de extracción de características distintos: *Análisis de componentes principales (PCA)* y *Non-negative matrix factorization with sparsity constraints (NMFSC)*.
- Utilización de *múltiples canales de información* y combinación de los resultados en función del objeto a detectar.
- Uso de *escalas o resoluciones distintas* en la escena global para facilitar la búsqueda en entornos complejos donde se desconoce la distancia a la que se encuentra el objeto de interés.
- Adaptación del detector al objeto de interés mediante la ponderación adecuada de los canales de información utilizados.
- Ajuste automático del umbral de detección en función del objeto.

Nuestro método de detección visual de objetos está compuesto por la combinación de dos etapas de búsqueda: en una primera etapa, que denominamos *detección multicanal PCA*, se utiliza la información de las imágenes de entrenamiento en varios canales comprimida con PCA, con el objetivo de predecir zonas dentro de la escena donde la probabilidad de que esté el objeto es mayor y, paralelamente, en otra etapa, de *detección multiescala NMFSC*, utilizar la información de las imágenes de entrenamiento comprimidas con NMFSC, a distintas escalas o resoluciones, para localizar el objeto de interés en la escena global. Finalmente, la combinación de ambas etapas permite delimitar la región de la escena en la que se encuentra el objeto.

2 DESCRIPCIÓN DEL DETECTOR

La estructura del algoritmo de detección de objetos propuesto se resume en las figuras 2, 3 y 4, donde las dos etapas antes señaladas se han subdividido para obtener un más alto nivel de detalle del algoritmo. A continuación se explican los contenidos de las mencionadas figuras.

En primer lugar, el gráfico de la figura 2 ilustra el proceso de detección de objetos propuesto de manera global. Como ya se ha dicho anteriormente,

nuestro método de detección de objetos está compuesto por la combinación de los resultados obtenidos en dos etapas o algoritmos independientes: *detección multicanal PCA* (algoritmo de detección que utiliza como representación visual del objeto la información de las imágenes de varias vistas del objeto, en diversos canales, mediante su compresión con PCA) y *detección multiescala NMFSC* (algoritmo de detección que emplea como representación visual del objeto la información de las imágenes de entrenamiento del objeto, en el canal ON-OFF [8], comprimidas con NMFSC).

Dichos algoritmos se han esquematizado en las figuras 3 y 4 separando los procesos *off-line* y *on-line* que ambos incluyen. Se utilizará esta división para una descripción más comprensible de su funcionamiento.

2.1 Procesos *off-line*

Los procesos a ejecutar *off-line* son aquellos pasos que se efectúan durante el entrenamiento del detector para un objeto de interés en concreto. Dado un cierto objeto, se capturan una serie de imágenes de entrenamiento que muestren todas las posibles vistas diferentes del objeto. En la figura 3 aparece un conjunto de 12 imágenes de entrenamiento para un objeto de interés, *la caja amarilla*. A partir de estas imágenes de entrenamiento se construyen dos modelos del objeto: un modelo basado en la apariencia visual, generado utilizando *fragmentos* de las imágenes y reducido con PCA, que se empleará durante la etapa de detección multicanal, y un modelo de apariencia a partir de las imágenes de entrenamiento completas, reducido con NMFSC, que se utilizará en la etapa de detección multiescala.

El conjunto de imágenes de entrenamiento forma el conjunto de información del objeto en el espacio imagen de dimensión igual al número de píxeles de la imagen, esto es, dadas N imágenes de entrenamiento de dimensión M , forman un conjunto de N vectores de dimensión M en el espacio imagen. Es posible reducir la dimensión de estos vectores mediante la técnica de PCA a través de la matriz de transformación \mathbf{U}^T compuesta por los vectores propios de la matriz de covarianzas de los datos centrados en el espacio original. Y ésta es la transformación que se aplica a un conjunto de fragmentos (ventanas o vistas parciales) de las imágenes de entrenamiento. Concretamente, utilizamos una arquitectura tipo subespacio objeto PCA [8], puesto que todas las imágenes pertenecen a la misma clase, para reducir la información de las imágenes de entrenamiento. El motivo principal de utilizar fragmentos de las imágenes de entrenamiento y no las imágenes completas para

la generación de los modelos PCA del objeto en la etapa de detección multicanal, es que los fragmentos o características de complejidad intermedia son óptimos para el reconocimiento visual de objetos [9].

La compresión de la información de los fragmentos de las imágenes de entrenamiento se realiza de forma múltiple para cada uno de los distintos canales de información, proceso denominado en la figura 3 como *obtención información multicanal*. Esto es, se utilizan las imágenes de entrenamiento filtradas en varios canales: imagen en niveles de gris, imagen en RGB, imagen en H, histogramas en diversos espacios de color, imagen de bordes, etc. Estos canales se describen ampliamente en [8].

La figura 5 ilustra esta etapa de obtención de los fragmentos (fig.5(a)) y el modelo PCA obtenido en un cierto canal de información a partir de la información de los fragmentos de las imágenes (fig. 5(b)). Cada uno de los puntos tridimensionales de la figura 5(b) es un fragmento de las imágenes de entrenamiento en un cierto canal, reducido a 3 dimensiones mediante la transformación PCA definida por los tres primeros vectores propios de la matriz de covarianzas de los datos. Este conjunto de puntos representa la información visual del objeto en el canal seleccionado, es decir, es el modelo de reconocimiento visual del objeto y se utilizará durante la detección para comprobar si las imágenes de test son semejantes o no a este modelo del objeto en dicho canal. La notación utilizada en la figura 3 para designar estos modelos del objeto en cada canal es la siguiente:

$$\{\mathbf{P}_1^*, \mathbf{P}_2^*, \dots, \mathbf{P}_i^*, \dots, \mathbf{P}_N^*\} \quad (1)$$

siendo i un canal de información dado, N el número total de canales y \mathbf{P}_i^* la matriz en cuyas columnas residen las proyecciones de los fragmentos (puntos tridimensionales en la figura 5(b)).

Se generarán tantos modelos PCA como canales de información se utilicen, siendo la dimensión de las proyecciones PCA un parámetro ajustable dentro del algoritmo, a mayor dimensión, menor compresión de la información pero menor error de reconstrucción de los datos [2].

Durante la etapa *off-line* de detección multiescala NMFSC también se genera un modelo del objeto en apariencia, \mathbf{P}^* , pero esta vez la transformación utilizada es NMFSC, y sólo en un canal de información, el doble canal *ON-OFF* [8].

También *off-line* y mediante aprendizaje automático se obtienen una serie de parámetros que se utilizarán durante las etapas *on-line* de ambos algoritmos: los *pesos* de los canales, que

ponderan la importancia de cada canal durante la detección del objeto, y el *umbral* de detección, que permite la segmentación del objeto en la escena.

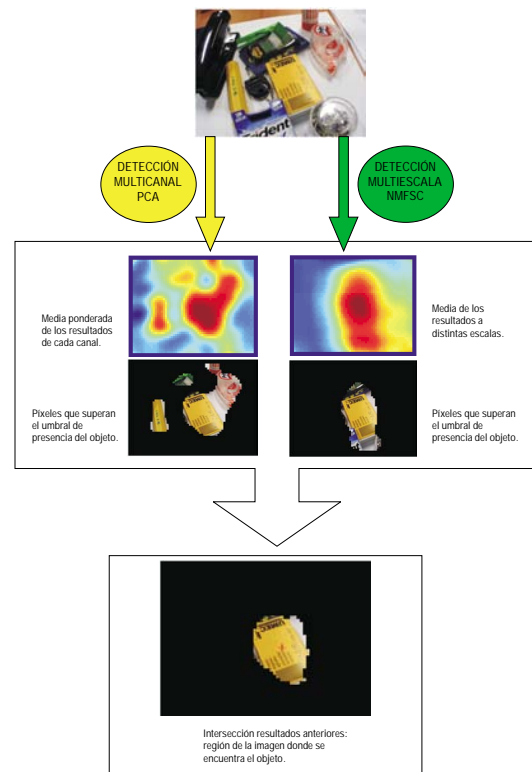


Figura 2: Estructura general del algoritmo de detección de objetos propuesto.

2.2 Procesos *on-line*

Los procesos *on-line* de ambas etapas permiten realizar la detección del objeto de interés en modo automático. Cada uno de ellos proporciona un mapa de probabilidad de la escena en función de la posible presencia del objeto de interés. La figura 2 muestra los mapas de probabilidad globales obtenidos tras aplicar cada una de estas etapas a la escena que se muestra en la parte superior de la figura. Debajo de cada mapa de probabilidad aparecen visibles las regiones de la escena original que superan un cierto umbral de probabilidad y que acotan la zona donde se encuentra el objeto de interés (*la caja amarilla*) permitiendo su segmentación respecto de la escena. En la parte inferior de la figura se muestra la combinación (*intersección*) de dichas etapas: se ha conseguido delimitar con mayor precisión la posición del objeto en la escena global.

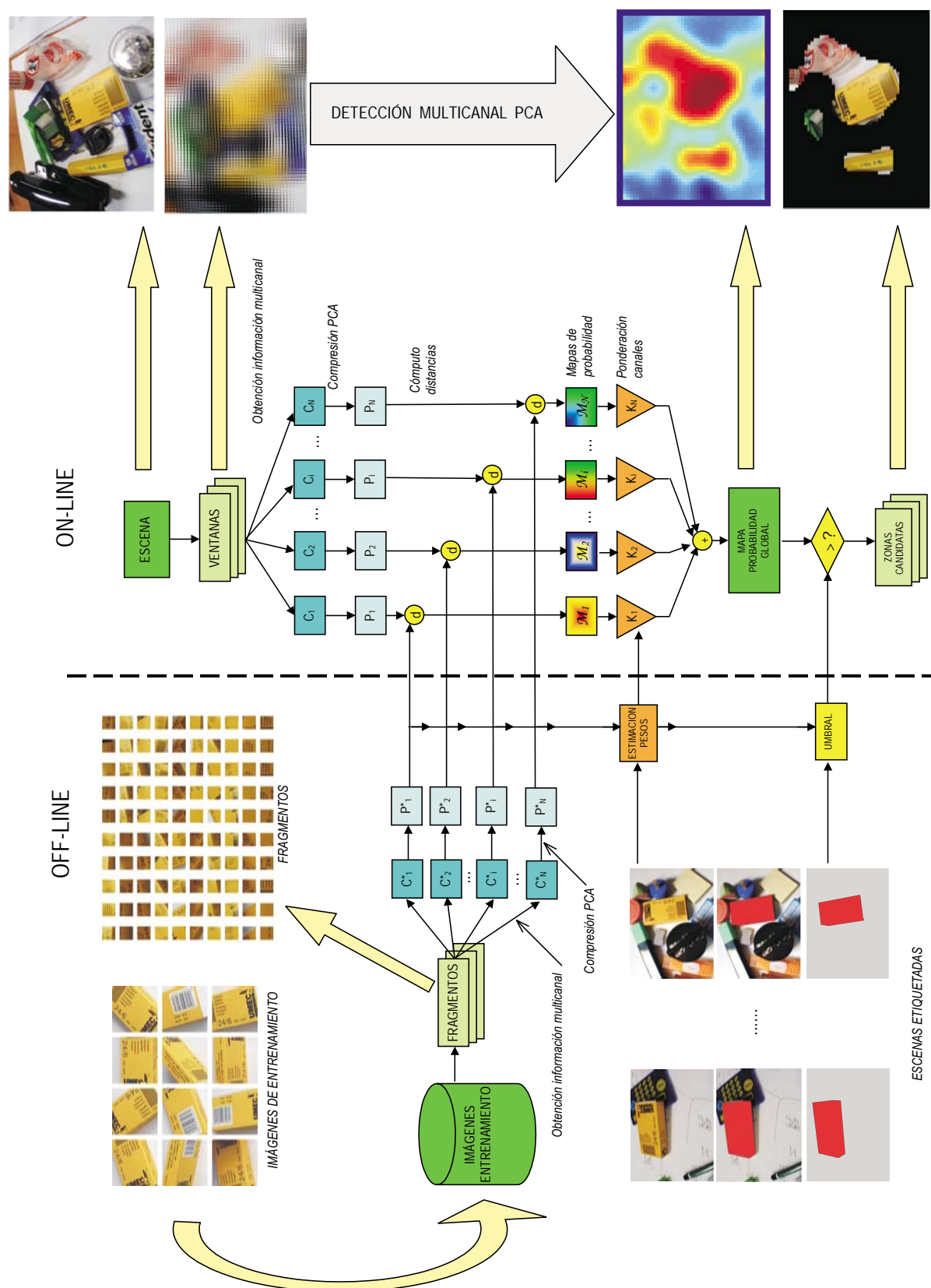


Figura 3: Estructura de la detección multicanal PCA.

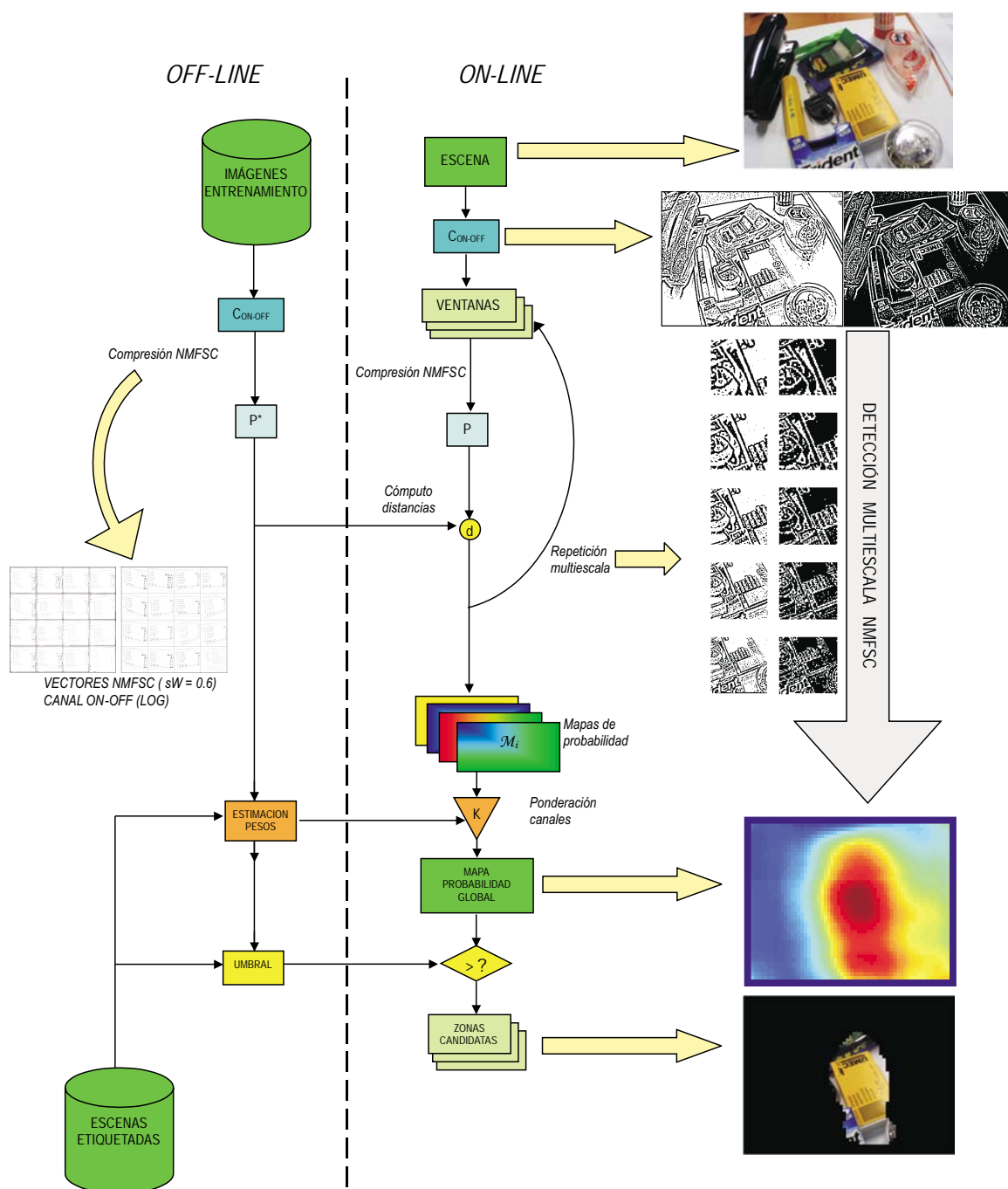


Figura 4: Estructura de la detección multiescala NMF.

La figura 3 ilustra la fase *on-line* de la detección multicanal PCA: la escena se divide en múltiples ventanas, \mathbf{v}_k , por cada uno de los canales de información utilizados (*obtención de información multicanal*, tal como se denomina en la figura 3), y estas ventanas son reducidas dimensionalmente mediante las matrices de transformación obtenidas durante el entrenamiento para cada canal i (*compresión PCA*). Siendo \mathbf{P}_i la matriz de las proyecciones de las ventanas de la escena en el canal i . Las proyecciones de las ventanas, \mathbf{P}_i , se comparan con los modelos obtenidos *off-line*, \mathbf{P}_i^* , utilizando una medida de distancia (*cómputo distancias*); en concreto, se ha utilizado la distancia Euclídea, que junto con la transformación PCA empleada equivale a la distancia de Mahalanobis, puesto que el método de PCA utilizado incluye el escalado posterior con los valores propios (1 proceso de *whitening*).

De este modo, por canales, a cada ventana \mathbf{v}_k se le asocia un valor igual a su distancia mínima al modelo, que será inversamente proporcional a la probabilidad de que el objeto de interés se encuentre en la ventana considerada:

$$(\mathbf{v}_k)_i \Leftarrow \min(\text{dist}[(\mathbf{p}_k)_i, \mathbf{P}_i^*]) \quad (2)$$

donde el índice i indica el canal de información, k es el índice de la ventana y \mathbf{p}_k es la proyección de la ventana centrada, $\hat{\mathbf{v}}_k$, en el subespacio PCA correspondiente al canal i :

$$(\mathbf{p}_k)_i = \tilde{\mathbf{\Lambda}}^{-1}(\tilde{\mathbf{U}}^T)_i \cdot (\hat{\mathbf{v}}_k)_i \quad (3)$$

En la figura 3 se emplea \mathbf{P}_i para denotar la matriz que reúne todas las proyecciones de las ventanas, $(\mathbf{p}_k)_i$, en el canal i considerado.

Según el número de veces que un mismo píxel aparece en varias ventanas, es posible asignarle un valor de distancia al modelo, realizando el promedio entre todas las distancias asociadas a él en cada una de las ventanas en la que es utilizado. Esta asignación entre píxeles de la escena y su distancia respecto al modelo del objeto depende del paso y el tamaño de la ventana utilizado, puesto que la fragmentación de la escena en múltiples ventanas y por tanto el número total de ventanas, N_v , depende de estos dos parámetros.

Continuando con la descripción de la etapa de detección multicanal PCA, mediante la asignación de distancias a cada píxel de la imagen, se consigue obtener una imagen en distancias inversamente proporcional a la probabilidad que el píxel pertenezca al objeto de interés. Esta imagen, matriz o mapa de distancias se denotará como \mathcal{D}_i . Con el fin de poder combinar los mapas de distancias obtenidos en cada uno de los distintos ca-

nales, es necesario realizar la normalización de dichos mapas de distancias. Este proceso de normalización puede realizarse según se describe en [8]. Al mapa de distancias normalizado se le denomina mapa de probabilidad, \mathcal{M}_i .

Tal como se aprecia en la figura 3, se obtienen tantos mapas de probabilidad del objeto en la escena, como canales de información se consideren. Posteriormente, se realiza una ponderación de estos mapas, para obtener un mapa de probabilidades global de la escena, \mathcal{M} , siendo la siguiente expresión la relación entre canales más sencilla a considerar:

$$\mathcal{M} = \frac{\sum_{i=1}^N K_i \cdot \mathcal{M}_i}{\sum_{i=1}^N K_i} \quad (4)$$

En la parte inferior derecha de figura 3 se muestra un ejemplo de mapa de probabilidades global. Las áreas en rojo, señalan las zonas de mayor probabilidad en las que puede encontrarse el objeto; mientras que las áreas en azul indican la ausencia del objeto en esas zonas de la escena. Este mapa es el resultado final de la etapa de detección multicanal PCA y acota las regiones en la escena donde puede hallarse el objeto.

De forma equivalente a la etapa de detección multicanal PCA, en la etapa *on-line* de detección multiescala NMFSC se utiliza el modelo del objeto \mathbf{P}_{NMFSC}^* para localizar la zona dentro de la escena en la que el objeto se encuentra con mayor probabilidad. En esta etapa sólo se utiliza un canal de información visual, el canal ON-OFF, pero en cambio, el barrido de la escena se repite varias veces con tamaños de ventanas distintos con el objetivo de obtener diversos mapas de probabilidad en función del tamaño de la ventana utilizado (o la operación equivalente: utilizar una ventana de tamaño fija, pero emplear escenas escaladas a distintos tamaños). Este hecho permitirá la localización del objeto aunque su posición respecto de la cámara no sea la misma que la utilizada durante la captura de las imágenes de entrenamiento. Este proceso de repetición de la búsqueda con tamaños de ventanas distintos se ha denominado en la figura 4 *repetición multiescala*. En dicha figura se muestra a modo de ejemplo un conjunto de 5 ventanas, en el canal ON-OFF, a distintas escalas, de forma que en cada par de ellas se aprecia como la escala de búsqueda del objeto en la escena es diferente.

Paralelamente a la etapa previa, los mapas de probabilidad obtenidos en cada una de las diferentes escalas de búsqueda se combinan finalmente para obtener un mapa de probabilidad global de la escena mediante la ecuación 4. En el proceso de combinación de resultados de la búsqueda en

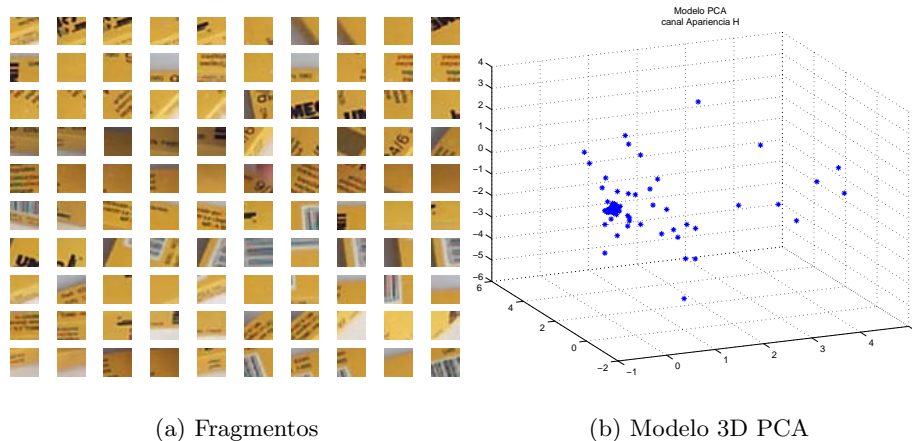


Figura 5: Los fragmentos obtenidos a partir de las imágenes de entrenamiento del objeto se comprimen utilizando PCA.

múltiples escalas, el valor de los pesos de los canales a distintas escalas depende de la aplicación. Por ejemplo, si se desconoce la distancia del objeto a la cámara, todos los canales se ponderan de igual forma, como podría ser el caso en una aplicación de navegación robótica, en la que el robot en principio debe localizarse dentro de una zona determinada. Sin embargo, si la distancia del objeto a la cámara se conoce, como sería el caso, por ejemplo, de la detección de objetos que circulan sobre una cinta transportadora delante de un robot fijo en una posición de la planta, se daría un peso mayor a las escalas que permiten detectar mejor el objeto, pudiendo incluso eliminar de la búsqueda algunas escalas muy lejanas a la posición esperada del objeto.

Este mapa de probabilidad global es el resultado final de la etapa de detección multiescala NMFSC y junto con el mapa de probabilidad global de la etapa de detección multicanal PCA servirá para determinar con mayor robustez y exactitud la posición del objeto y segmentarlo adecuadamente respecto del resto de la escena.

2.3 Combinación de las etapas multiescala y multicanal

Cada una de las etapas de detección proporcionan un mapa de probabilidad relativo a la escena que informa sobre la mayor o menor probabilidad de que un determinado píxel pertenezca o no al objeto de interés. Además, cada mapa de probabilidad lleva asociado un parámetro umbral que permite decidir si el objeto está o no presente en la escena, y en caso positivo, determinar en qué zona en concreto.

Luego, asumiendo que cada mapa de probabilidad

lleva asociado un umbral que permite decidir sobre la localización del objeto en la escena, cada una de las etapas producirá una escena segmentada en la que se visualizará el área correspondiente al objeto, tal como se muestra en las dos imágenes umbralizadas de la figura 2.

El aumento o disminución de cada uno de los umbrales en sus respectivos mapas de probabilidad produce que el área dentro de la escena asociada al objeto sea mayor o menor, con la consiguiente inclusión de más falsos positivos o bien el aumento de falsos negativos.

Nuestra propuesta consiste en la combinación de las escenas umbralizadas de ambas etapas de forma que disminuyan los falsos positivos de cada una de las dos etapas por separado. Esta combinación consiste, básicamente, en realizar la intersección entre los píxeles de ambas escenas segmentadas o umbralizadas, con el objetivo de obtener mayor certeza de la presencia del objeto en los píxeles resultantes, puesto que serán píxeles positivos de pertenencia al objeto, proporcionados por dos métodos de compresión muy diferentes. Esta forma de combinar los resultados de ambas etapas requiere que los umbrales utilizados sean lo suficientemente permisivos, de forma que se eviten en la medida de lo posible los falsos negativos. La figura 2 muestra en su parte inferior el resultado de realizar esta intersección de píxeles entre las dos escenas umbralizadas en cada una de las etapas. Se puede apreciar como se ha reducido claramente el número de falsos positivos. A partir de esta escena final umbralizada se podrá obtener el centro del objeto como el centro de masas de la zona segmentada.

3 AJUSTE AUTOMÁTICO DE LA INFLUENCIA DE CADA CANAL

Los coeficientes K_i deben ser proporcionales a la fiabilidad de cada canal. Existen múltiples factores que afectan a tales fiabilidades relativas. Para ciertos objetos serán más fiables unos canales que otros. Por ejemplo, objetos con colores muy uniformes pueden ser distinguidos fácilmente a través de los canales de histogramas, mientras que objetos en los que los colores son más variables requieren que los pesos de los canales de color sean bajos para obtener buenos porcentajes de reconocimiento. Objetos con símbolos o texto podrán ser reconocidos más fácilmente utilizando los canales de forma. También, en función del tipo de entorno, puede variar la influencia relativa de cada canal. Fondos sencillos o incluso uniformes favorecen en general a todos los canales pero en particular a los basados en bordes. Dependiendo del tipo de uniformidad del fondo, se beneficia a ciertos canales: fondos uniformes en color benefician a los canales de histogramas; mientras que fondos uniformes en textura pueden beneficiar a otros canales. La existencia de todos estos factores hace que el ajuste de pesos óptimo varíe en función de cada experimento. En cualquier caso, el factor más relevante es el tipo de objeto a detectar, con lo que proponemos utilizar una ponderación específica de canales para cada objeto.

3.1 Propuesta de aprendizaje de pesos mediante entrenamiento

Se propone asignar los pesos de cada canal mediante un procedimiento de aprendizaje a partir de ejemplos de entrenamiento. Los ejemplos son escenas similares a las que el sistema encontrará durante el funcionamiento automático, en las que el objeto de interés se encuentra segmentado y etiquetado.

En función de la variabilidad previsible en las imágenes que encontrará el sistema en funcionamiento automático, los ejemplos de entrenamiento deberán cubrir en la medida de lo posible tal variabilidad (diversas condiciones de iluminación, distintos fondos, distintas distancias entre el objeto de interés y la cámara, etc.)

La idea básica consiste en considerar que la influencia de cada canal ha de ser proporcional a la correlación que existe entre el mapa de probabilidad de dicho canal y la imagen de la escena etiquetada (en la que las zonas donde se encuentra el objeto tienen valor 1 y el fondo de la imagen tiene valor 0). Para calcular correctamente la correlación, los mapas de probabilidad de cada

canal han de estar normalizados, es decir, el píxel de máxima distancia al modelo tendrá un valor de probabilidad 0 y el píxel de mínima distancia, un valor 1.

En resumen, el procedimiento a utilizar para calcular la influencia óptima de cada canal es el siguiente:

Para cada objeto:

1. Recopilar escenas de entrenamiento y etiquetarlas.
2. Calcular mapas de distancia, \mathcal{D}_i , en cada uno de los canales y escalarlos entre 0 y 1 linealmente (mapas de probabilidad, \mathcal{M}_i).
3. Calcular la correlación para cada escena y cada canal; a tales correlaciones se les llamará c_{ij} donde j hace referencia a la escena (se supone un total de T escenas) e i , al canal (se supone un total de N canales).
4. Calcular la fiabilidad o el peso de cada canal como la media de las correlaciones entre todas las escenas de entrenamiento:

$$K_i = \frac{1}{T} \sum_{j=1}^T c_{ij} \quad (5)$$

4 UMBRAL AUTOMÁTICO DE RECONOCIMIENTO

El resultado de los dos algoritmos de detección de objetos propuestos se representa mediante un mapa de probabilidades, que indica, para cada píxel de la imagen, la probabilidad de que el objeto a detectar ocupe tal píxel. En general, la probabilidad será tanto más cercana a la unidad cuanto más similar sea la escena a alguna de las imágenes (o los fragmentos) de entrenamiento. Se puede considerar que los valores más altos de probabilidad dentro del rango indican la presencia del objeto, aunque estos valores se encuentren alejados de la unidad. Si el único objetivo es detectar la posición del objeto en la imagen, basta con marcar como posición del mismo el píxel de mayor probabilidad (indicado con una cruz en la figura). Sin embargo, esta estrategia presenta algunos inconvenientes: no permite obtener una segmentación, aunque sea aproximada, del objeto y tampoco permite determinar si el objeto está presente o no en la escena.

Estos inconvenientes se pueden solventar con el

establecimiento de un umbral, de modo que todos los píxeles con valor de probabilidad superior al umbral sean considerados pertenecientes al objeto a detectar. Si ninguno de los píxeles supera el umbral se concluye que el objeto buscado no está presente en la escena.

Se han propuesto dos metodologías para el establecimiento de un umbral automático: *a través de la distribución espacial de los modelos de apariencia y mediante aprendizaje con escenas etiquetadas*. Ambos algoritmos se describen en profundidad en [8].

En la etapa de detección multiescala NMFSC, sólo se obtiene un umbral automático, ν_{ON-OFF} , puesto que sólo se dispone del modelo de apariencia NMFSC en el canal ON-OFF.

Por otro lado, en la etapa de detección multicanal PCA se obtiene un umbral, ν_i , para cada uno de los diferentes canales i . Sin embargo, no se utilizarán directamente sobre cada uno de los mapas de probabilidad de cada canal sino que se combinarán para obtener un umbral global, ν_G , que sea el responsable de la segmentación final del objeto mediante su uso sobre el mapa de probabilidad global. Este umbral se obtendrá a través de una expresión equivalente a la utilizada para el cálculo del mapa de probabilidad global:

$$\nu_G = \frac{K_1 \cdot \nu_1 + K_2 \cdot \nu_2 + \dots + K_N \cdot \nu_N}{K_1 + K_2 + \dots + K_N} \quad (6)$$

donde los coeficientes K_i son los pesos de los canales obtenidos mediante el procedimiento descrito en el apartado 3.

El primer método (basado en las distribuciones espaciales de los modelos) proporciona un umbral que es una distancia, luego antes de utilizar la expresión 6 para el cálculo de ν_G se debe normalizar el umbral de la misma forma que se normalizaban los mapas de distancia \mathcal{D}_i en mapas de probabilidad \mathbf{M}_i .

En el segundo método propuesto para la obtención del umbral (mediante aprendizaje con escenas etiquetadas), éste se obtiene como un valor de probabilidad puesto que se trabaja con los mapas de probabilidad normalizados de cada canal, \mathcal{M}_i , por lo que la expresión 6 es aplicable directamente una vez son calculados los umbrales de cada canal.

5 Validación del método propuesto

Se ha desarrollado un entorno de pruebas para el estudio y validación del algoritmo de detección de objetos propuesto. Todas las aplicaciones desarrolladas se han realizado sobre *Matlab* para facilitar la visualización de todas las etapas del proceso de

detección. Las imágenes de los objetos, tanto las de entrenamiento como las de test, pueden capturarse utilizando una sencilla webcam.

Básicamente, las aplicaciones desarrolladas se pueden clasificar en tres tipos: entorno de entrenamiento, entorno de experimentación y entorno de funcionamiento automático.

El entorno de entrenamiento permite realizar las siguientes operaciones descritas en secciones anteriores: obtención de fragmentos, creación de imágenes etiquetadas (escenas complejas etiquetadas), generación de los modelos de apariencia multicanal (PCA), generación del modelo de apariencia (NMFSC) para la etapa de detección multiescala, ajuste de los pesos de los canales automático y cálculo de los umbrales para ambas etapas con los dos métodos propuestos (basado en modelos y basado en escenas etiquetadas).

En el entorno de experimentación, con el objetivo de comprobar el funcionamiento adecuado de cada canal y el efecto de aplicar los umbrales sobre los mapas de probabilidad de la escena, se han desarrollado dos aplicaciones: la primera de ellas, denominada *visualiza mapas*, permite la visualización de todos los mapas de probabilidad, con la posibilidad de realizar o no los canales según el factor de escalado configurable γ y la otra aplicación, *visualizador interactivo*, permite tanto la visualización de una selección de mapas de probabilidad, como el ajuste manual de los pesos y umbrales. El aspecto del *visualizador interactivo* se muestra en la figura 6. En la parte izquierda del mismo, se muestran los mapas de probabilidad de diversos canales (se puede visualizar un máximo de 8 canales). En la parte central, se muestran, de arriba a abajo, la escena original, la suma ponderada de todos los mapas de probabilidad, y el conjunto de píxeles que superan un cierto umbral (píxeles donde es esperable que se encuentre el objeto de interés). Adicionalmente, sobre la escena original se marca una cruz en el punto de máxima probabilidad, de acuerdo con la suma ponderada de todos los mapas.

El entorno de funcionamiento automático permite realizar la detección del objeto de interés, sobre una imagen de entrada de la escena, a partir de la información generada y almacenada en el sistema durante el entrenamiento (modelos de apariencia, pesos de los canales, K_i y valores de los umbrales globales ν_G).

Agradecimientos

Este trabajo ha sido subvencionado por la Consejería de Empresa, Universidad y Ciencia a través del proyecto GV06/123, *Aprendizaje en teleoperación para la imitación del agarre humano me-*

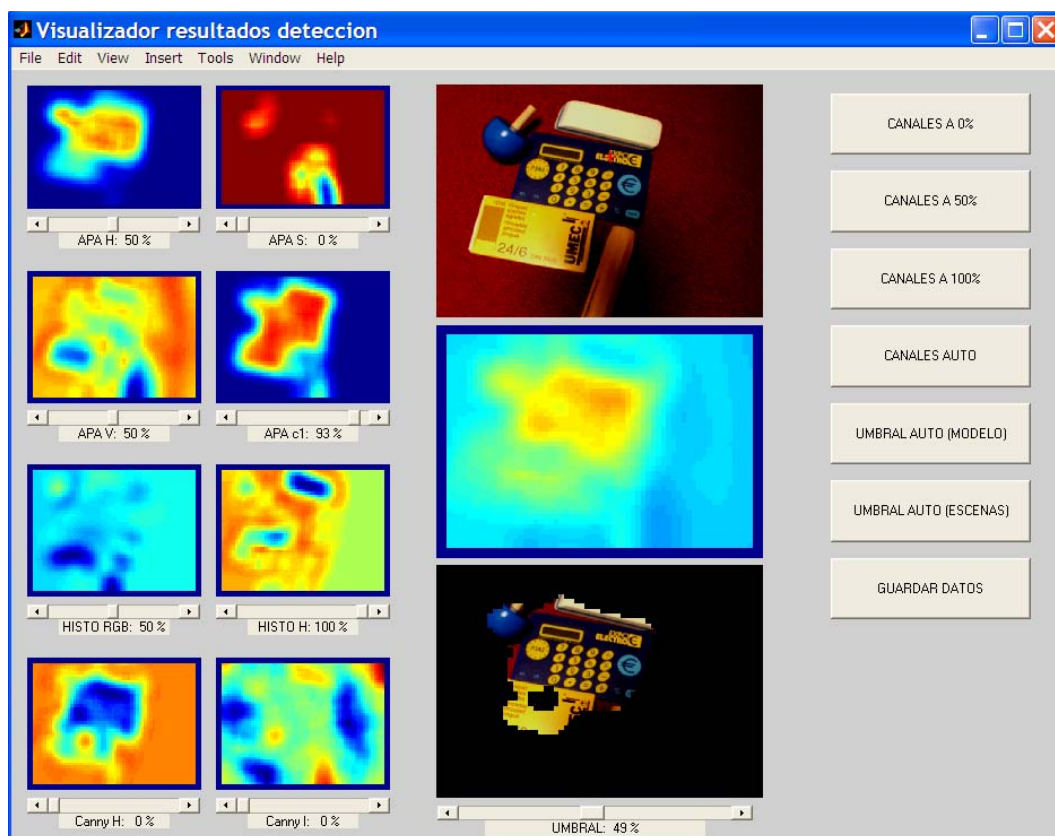


Figura 6: Entorno *visualizador interactivo*, objeto a detectar: la *caja amarilla*.

diante robots.

Referencias

- [1] Brunelli, R. and Poggio, T., (1993), Face recognition: features versus templates, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.15 , n.10, pp. 1042-1052.
- [2] Kirby, M., (2001), Geometric Data Analysis, Wiley Interscience.
- [3] Ohba, K. and Ikeuchi, K., (1997), Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19 , n.9, pp. 1043-1048.
- [4] Rajagopalan, A. N. and Burlina, P. and Chellappa, R., (1999) Higher Order Statistical Learning for Vehicle Detection in Images, ICCV.
- [5] Rowley, H.A. and Baluja, S. and Kanade, T., (1998), Neural Network-Based Face Detection, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.20 , n.1, pp 23-38.
- [6] Schneiderman, H., (2004) Feature-Centric Evaluation for Efficient Cascaded Object Detection, IEEE Conf. on Computer Vision and Pattern Recognition.
- [7] Sung, K.K. and Poggio, T., (1998), Example-Based Learning for View-Based Human Face Detection, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.20 , n.1, pp 39-51.
- [8] Vicente Ripoll, M.A, (2006), Reconocimiento y búsqueda selectiva de objetos en entornos no estructurados mediante apariencia multi-canal y multiescala, Tesis doctoral UMH.
- [9] Ullman, S. and Vidal-Naquet, (2002), M. and Sali, E., Visual features of intermediate complexity and their use in classification, Nature Neuroscience, vol.5 , n.7, pp 682-687.
- [10] Yang, M.H. and Roth, D. and Ahuja, N., (2000), SNoW-Based Face Detector, Advances in Neural Information Processing Systems 12, pp. 851-861.

INSPECCIÓN DE TUBERÍAS Y CABLES MEDIANTE VEHÍCULOS AUTÓNOMOS SUBMARINOS

Oscar Calvo Ibáñez
oscar@galiota.uib.es

Gerardo Gabriel Acosta
gerardo.acosta@ieee.org

Hugo Javier Curti
hcurti@galiota.uib.es

Resumen

En este trabajo se presenta el diseño y posterior desarrollo de la arquitectura de software para la navegación autónoma de un vehículo submarino. El módulo de generación de las trayectorias deseadas es la parte central del artículo, que también describe las tareas llevadas a cabo en el marco de los proyectos europeos AUTOTRACKER y AUVI para la inspección de oleoductos y cables. Se presenta el módulo de navegación desde su forma inicial – una máquina de estados – hasta su actual estado de implantación, basado en un sistema experto. El vehículo experimental es lo suficientemente robusto como para permitir inspecciones en profundidades de varios miles de metros. Se realizaron pruebas exitosas en el Mar del Norte, con lo que se demuestra que la tecnología actual está lo suficientemente madura para enfrentar la inspección autónoma en aguas profundas, prácticamente sin intervención humana. Los resultados de estas pruebas son también discutidos.

Palabras Clave: Robots Móviles Autónomos. Planificación de trayectorias. Sistema Basado en Conocimiento. Oleoductos submarinos.

1 INTRODUCCIÓN

Una de las aplicaciones más útiles de los vehículos submarinos autónomos (AUV) es el seguimiento de tuberías y cables sumergidos, principalmente con propósitos de mantenimiento. Fundamentalmente para prevenir una pérdida económica a la vez que preservar el ecosistema marino. El resultado es un interés cada vez mayor por la explotación comercial de tales inspecciones submarinas.

Actualmente se emplean vehículos operados por control remoto (ROV). Sin embargo esta aproximación posee dos limitaciones básicas en las grandes profundidades: la calidad de los datos

adquiridos es pobre debido a la fuerte perturbación del cordón de control remoto, y el costo asociado a cada misión, que requiere al menos una embarcación de superficie de gran envergadura, con el costo de su tripulación asociado. Adicionalmente, la velocidad de inspección es lenta (de alrededor de tres cuartos de nudo por segundo). Desventajas éstas que se ven acentuadas al incrementar las profundidades. La industria del petróleo, un caso de aplicación paradigmático, cada vez se mueve a mayores profundidades a medida que los recursos superficiales van agotándose.

En contraste, los AUV permiten una navegación más suave en las profundidades, y por ende, mejor calidad de datos adquiridos, ya que no tienen ningún cable hacia una plataforma de superficie. Este soporte de superficie necesario puede ser a su vez mucho más pequeño. En efecto, si el AUV puede alcanzar posiciones definidas en coordenadas globales y navegar en forma autónoma con un error de posición pequeño, y es capaz de seguir la lectura de unos sensores dedicados, las limitaciones se reducen a soportar las grandes presiones de miles de metros de profundidad y al almacenamiento de energía en baterías de a bordo. Durante los últimos años se han realizado pruebas en robótica submarina para el seguimiento de cables y tuberías con diferentes grados de éxito, como por ejemplo el Twin-burger 2, guiado por imágenes provenientes de cámaras [1], aunque para aguas oscuras es preferible emplear sonares o una combinación de varios sensores como en RAIS [2]. El proyecto AUTOTRACKER (GRD1-2000-25150) financiado por la Unión Europea fue propuesto para mostrar que con la tecnología actual se puede afrontar el desafío de explorar a grandes profundidades una estructura humana en forma autónoma y generar un prototipo listo para ser transferido para explotación comercial [3]. En la Figura 1, puede apreciarse la arquitectura del prototipo AUTOTRACKER en un diagrama de bloques.

El módulo de **navegación** está formado por un sistema de posicionamiento geográfico (GPS) que da

una posición absoluta en coordenadas globales como una dupla (latitud, longitud), y un sistema de navegación inercial basado en giróscopos que da la posición relativa a la última medida del GPS una vez que el vehículo se sumerge. Estos sistemas vienen provistos en el submarino Geosub empleado en las pruebas (Figura 2). El módulo de navegación hace un “broadcast” de la lectura de estos sensores al resto de los módulos que requieren de la posición.

Para la adquisición de los datos de inspección se empleó un sonar multi haz (MBE). Los datos crudos provistos por este sonar son procesados por el módulo “*MBE Tracer*”, que predice una trayectoria para el objetivo empleando técnicas de procesamiento de imágenes [4]. Cuando se construye la instalación sumbarina se archivan datos relativos a posiciones, profundidades y corredores, en archivos históricos o “*legacy data*”, que fueron puestos en formato disponible para funcionamiento de los módulos de fusión de sensores y el planificador de misión dinámica que se informa en este artículo. Dicho formato **ADF** (*autotracker description files*)

comprenden una descripción del fondo marino (BSF) y del objetivo a inspeccionar (PSF). La trayectoria del oleoducto es estimada a partir de los sensores especiales que comprenden el MBE mencionado anteriormente, un sensor magnético (MAG) y los archivos ADF con información a priori de la posición de la tubería. Toda esta información es combinada por un **módulo fusor de sensores** (SFM), dando como salida la estimación de la posición y trayectoria del objetivo a inspeccionar, conjuntamente con una medida del error de esa estimación. Usando esta información del módulo de fusión de sensores y la información a priori, el módulo replanificador de misión, llamado **ATModule**, decidirá la trayectoria a seguir en las diferentes situaciones de búsqueda, seguimiento, readquisición, y otras. Dichas trayectorias están formadas por cuatro (4) puntos de paso (*waypoints*) que deben ser alcanzados por el vehículo. ATModule es el desarrollo principal presentado en este artículo, construido también por el equipo de la UIB, y será descrito en profundidad en las siguientes secciones.

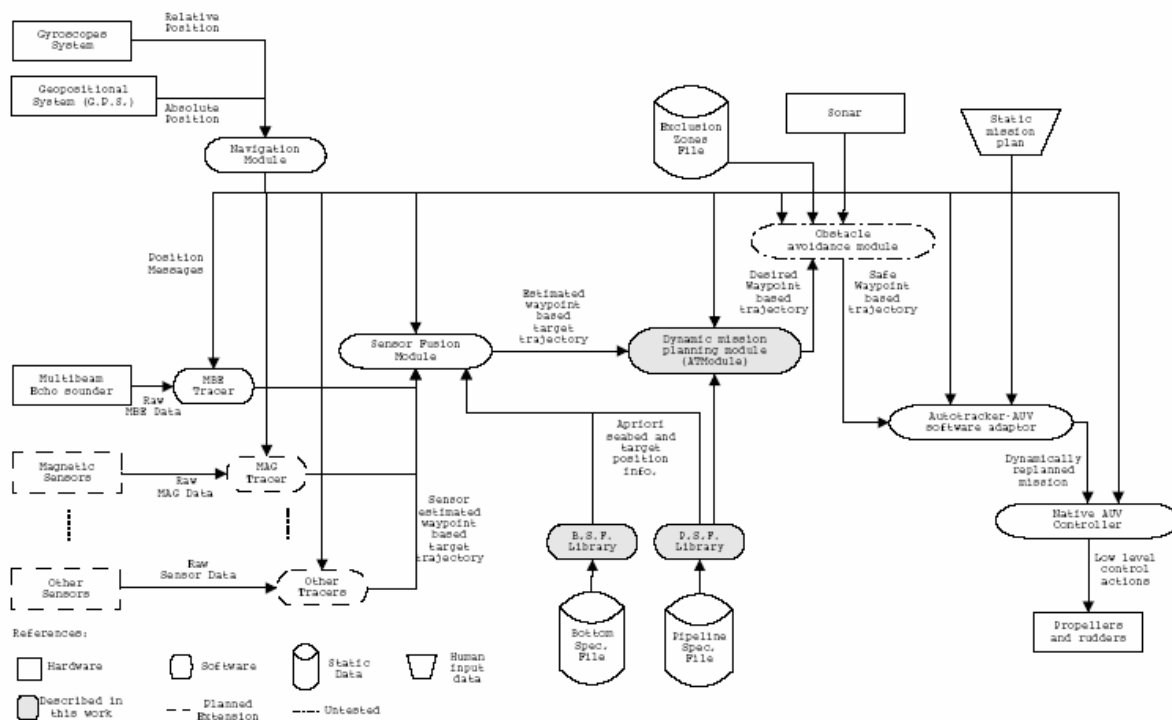


Figura 1: Diagrama en bloques de los módulos de AUTOTRACKER.

El **sistema de detección de obstáculos** (OAS) toma las trayectorias propuestas por el módulo anterior (ATModule) y las valida usando datos provenientes de un sonar frontal y de las zonas de exclusión, también a partir de los legacy data. En caso de aparecer un obstáculo, modificará las trayectorias propuestas. Al momento de las pruebas reportadas aquí, este módulo estaba en desarrollo [5] y se empleó el provisto dentro del Geosub, solamente con propósitos de seguridad del vehículo.

De este modo, las trayectorias validadas provenientes del OAS son empleadas por el **módulo adaptador AUTOTRACKER-AUV** para cambiar dinámicamente el plan de misión estático que originalmente propuso el operario humano antes de sumergir el vehículo para realizar una inspección. Esta arquitectura simplifica enormemente la tarea de portabilidad del software a cualquier otra plataforma de AUV, como se verá luego para el caso del vehículo AUVI. A partir de las trayectorias de 4

puntos de paso, los módulos de guiado y control, nativos del AUV, generan las acciones de control para timones y propulsores.

Los módulos de la Figura 1 fueron integrados gradualmente a lo largo de casi tres años de trabajo y probados primero en simulaciones de computadora [6] y luego experimentalmente en pruebas en el mar del Norte. En las secciones que siguen se presentan las consideraciones de diseño y el desarrollo del módulo de generación de trayectorias deseadas (ATModule), conjuntamente con los resultados de las pruebas mencionadas. Asimismo se presenta el prototipo AUVI, una versión simplificada de muy bajo costo del vehículo AUTOTRACKER que se está desarrollando con fines de I+D.

2 METODOLOGÍA

Cuatro hipótesis de trabajo fueron tomadas como punto de partida en este sentido: 1. el sistema de control del AUV era lo suficientemente robusto como para recibir las coordenadas absolutas de latitud, longitud y profundidad, y posicionarse con un error acotado; 2. la trayectoria deseada (salida del ATModule) sería la entrada del sistema de evasión de obstáculos (OAS) que daría la trayectoria posible al sistema de control del AUV; 3. cada trayectoria era un conjunto de cuatro puntos de paso en coordenadas absolutas en la superficie terrestre, incluyendo la profundidad, velocidad y modo en el que debía alcanzarse cada punto de paso; 4. la posición del objetivo a inspeccionar sería estimada por el módulo de fusión de sensores (SFM) con un error asociado (S) en tres coordenadas (x,y,z) relativas al AUV.



Figura 2: AUV provisto por Subsea7 para las pruebas.

2.2 APROXIMACIÓN EN MÁQUINA DE ESTADOS (ME)

Se construyeron tres máquinas de estados: *Main Machine*, *Operate Machine*, y *Search Machine*.

Main Machine tiene dos estados: de operación y de vacío. En el estado de vacío (Idle state), el módulo ATModule corre en modo de mantenerse aparte. Se mantiene así hasta que se recibe un mensaje de operar, en cuyo caso se pasa al estado de operar. La máquina permanece así hasta que se concluye la misión o se recibe un mensaje de *standby*, retornando al estado de vacío. Todos estos mensajes con el formato del marco propuesto por el equipo de la HWU. La Operate Machine siempre empieza en el estado de encontrar el inicio de la misión (*FindStart*). Una vez que este punto de paso inicial se encuentra, el siguiente estado es buscar. Dentro de este estado, se arranca la Search Machine o máquina buscar. En esta situación, la máquina buscar genera una trayectoria en forma de cortadora de césped de modo tal de permitir a los sensores realizar una completa revisión del área donde se supone de antemano que estará el oleoducto para iniciar la inspección. Si efectivamente se encontró al oleoducto, la máquina operar va hacia el estado *BackToStart* en el cual el AUV se posiciona en el punto de inicio con rumbo paralelo a la dirección supuesta del objetivo, y a la altura y desplazamiento lateral predefinidos en la especificación de misión estática. Luego pasa al estado de seguimiento (*Track*) dentro de la máquina operar, en el que se generan trayectorias paralelas al ducto, hasta que se alcanza el final de misión o los sensores no detectan más el oleoducto. Cada vez que se pierde el objetivo, se inicia una nueva búsqueda en la vecindad del punto en el que se considera perdido. Si aun esta nueva búsqueda es infructuosa, la máquina operar va al estado salto (*Skip*) en el que se genera una trayectoria con una dirección estimada a partir de los legacy data y con una longitud que se establece de antemano también en la especificación estática de la misión. En las nuevas coordenadas que se supone están también dentro del corredor, se pasa el control a la máquina buscar. Ésta iniciará una nueva búsqueda. Si nuevamente se fracasa en encontrar al objetivo, el procedimiento de saltar y buscar se repite un número finito de veces, también especificado en la planificación de la misión. Por su parte, cada estado tiene un chequeo del tiempo, y cada vez que se produce la extinción de un tiempo límite, cada máquina va a un estado especial (*Timeout*) que concluirá la misión con un mensaje de error. Además, cuando se alcanza un número máximo de búsquedas fracasadas, la máquina buscar también va a un estado de error, con las mismas consecuencias que el estado de tiempo agotado.

La Search Machine arranca en el estado primera mitad (*FirstHalf*) dentro del cual el AUV es guiado desde el punto de inicio de búsqueda hasta el lado izquierdo del corredor visto desde arriba, girando en sentido antihorario un ángulo predefinido (?), siempre tomando la supuesta dirección del objetivo como referencia. Este ángulo, de aproximadamente 60° es para una mejor detección por parte del sonar

multihaz. Luego pasa a *Line* y *LastHalf* obteniendo un patrón de trayectoria del tipo cortadora de césped. Si dibujando este patrón, el objetivo es encontrado en un punto, en cualquiera de los tres estados (*FirstHalf*, *Line*, *LastHalf*), éste es abandonado con un flag de éxito. Esta máquina de estados fue desarrollada en lenguaje C++ estándar, bajo SO Linux, que es el que corre en el AUV en el ambiente real.

2.3 APROXIMACIÓN POR SISTEMA EXPERTO (SE)

Cuando se realiza una inspección suelen aparecer situaciones complejas, como la presencia de redes de pesca o anclas sobre la tubería, o decidir cuál es la tubería a seguir cuando se cruzan dos o están muy próximas, o el objetivo aparece enterrado intermitentemente. La máquina de estados puede no prever tales situaciones y dar una salida en error. Como el objetivo final es proveer al AUV de la “inteligencia suficiente” como para enfrentarse a situaciones reales imprevistas en el mundo submarino, se recurrió a incorporar la experiencia humana en el desarrollo de las tareas de inspección. De este modo, se aprovechó parte del conocimiento de los operarios de vehículos de control remoto (ROV) que rutinariamente realizan inspecciones como las que se le exigirán al vehículo autónomo. Este conocimiento fue elicited y codificado en forma de un sistema experto, llamado *EN4AUV* (*Expert Navigator for Autonomous Underwater Vehicles*) [7], construido empleando CLIPS, una herramienta de desarrollo de sistemas expertos basada en C. Su principal característica es la de determinar la situación actual para actuar consecuentemente, en un comportamiento reactivo, claramente guiado por datos. Así, EN4AUV es un sistema experto reactivo que toma la decisión (adecuada) de actuar para cada diferente situación, considerando el estado en que se encuentra el oleoducto, el tipo de misión con sus diferentes parámetros, y la historia de inspección previa de la misión actual para poder replanificar. Para determinar la situación se emplean ciertas variables monitorizadas que se organizan en una clase denominada escenario, a cuyas instancias de trabajo presente acceden cerca de 50 reglas que conforman la base de conocimiento actual. A medida que se adquiere mayor experiencia frente a nuevas situaciones, nuevos escenarios pueden añadirse para completar y actualizar la base, dando como resultado su crecimiento incremental. Los escenarios se basan fundamentalmente en dos ideas: los tipos de inspección y el estado de seguimiento. El tipo de inspección se define antes de iniciar la misión y establece si se adquirirán datos de uno o más oleoductos, la profundidad y velocidad de navegación, el ancho del corredor, los fallos en readquisición tolerados, y otros.

El otro componente básico de la determinación del escenario es el estado de seguimiento que cambia cuando el SFM actualiza sus sensores. A partir de esto, el EN4AUV puede decidir el estado de la tubería/cable (si está enterrado, expuesto, intermitente, tramo libre) y cómo está el AUV en lo que se refiere a la tubería/cable (si evita un obstáculo, con respecto al objeto en estudio considerado como encontrado o perdido, o retornando a una posición previa conocida). Por ejemplo, incluso cuando el robot está siguiendo la tubería que aparece enterrada y expuesta en forma intermitente, el ATModule debe ser “consciente” de la situación previa de manera de optimizar tiempos y no realizar una búsqueda en el caso que la tubería se pierda, porque pronto puede aparecer.

```
(defrule R05.1
  (R4SD)
  ?a <- (R4SD)
  ?ws <- (object (is-a WORKING_SCENARIO)
    (Count_Reacq ?cr)(Search_results ?sr) (Movie $?MOVIE)
    (Follow_status ?FS))
  ?sv <- (object (is-a SURVEY) (Max_Reacquire ?mx))
  (test (<= ?cr ?mx))
  (test (eq ?FS LOST))
  (test (eq (send [OBJ_STUDY] get-
    Present_Layout_Status) NOT_DETECTED_NOT_BURIED))
  test (= 0 ?sr))
=>
  (assert (Current_scenario SC8))
  (assert (PPLS notready))
  (retract ?ws)
  (insert$ $?MOVIE 1 SC8)
  (send ?ws put-Movie $?MOVIE)
  (retract ?a) ;to avoid the assertion of multiple scenarios in
  the same KB query
  (printout t "CLIPSMACHINE: R05.1 Current Scenario is
  SC8 -searching a pipe/trunkline" crlf)
  (printout t "CLIPSMACHINE: inserting movie" $?MOVIE
  crlf))
```

Figura 3: Una regla de la base de conocimiento para la determinación del escenario 8, en la típica sintaxis de CLIPS

Una vez establecidos los escenarios, EN4AUV debe generar una trayectoria deseada o decidir la readquisición de la tubería/cable. Para procurar una trayectoria deseada, las acciones se organizan en un conjunto de algunas pocas sub tareas simples: **búsqueda inicial** (*first search*), **búsqueda** (*search*), **vuelta al inicio** (*back to start*), **salto** (*skip*), y **seguimiento** (*track*). Entonces la trayectoria final del AUV se puede construir por una de estas sub tareas, o por una concatenación de ellas. Ejemplos de los 14 escenarios desarrollados para los ensayos son los siguientes:

1^{er} Escenario: El AUV está siguiendo una tubería expuesta, navegando sobre ella, a una distancia fija

menor o igual a 5 metros. El MBE y el MAG lo detectan.

2^{do} Escenario: El AUV está siguiendo una tubería enterrada, pasando sobre ella, a una distancia fija menor o igual a 5 metros. El MBE puede no ser capaz de detectarla, pero el MAG puede ubicarlo de todos modos.

3^{er} Escenario: El AUV está siguiendo una tubería intermitente (expuesto intermitentemente y enterrado) a una distancia fija. Ésta es una secuencia alternativa de los escenarios número uno y dos.

7^{mo} Escenario: El AUV está buscando una tubería enterrada. No existen lecturas del MBE, el MAG indicará la detección solo cuando el AUV esté exactamente sobre el tubo.

9^{no} Escenario: El AUV está buscando una tubería en presencia de uno o más tubos u otros objetos magnéticos en el área.

En la Figura 3 se muestra una de las reglas de la base de conocimiento, responsable de determinar la presencia del escenario 8.

3 PRUEBAS Y RESULTADOS

Todo el hardware del AUTOTRACKER está ubicado en el interior de un recipiente estanco de titanio que soporta grandes presiones, de hasta 3000m de profundidad. Dentro del recipiente hay cuatro computadoras industriales embebidas (del tipo PC-104), conectada a través de una red ethernet, corriendo con sistema operativo GNU/Linux. El sistema y las librerías base inicializan desde tarjetas flash. Una de ellas posee un disco rígido, y lo comparte con las demás utilizando el protocolo NFS. Ambas aproximaciones, máquina de estados y sistema experto, fueron chequeadas previamente en simulaciones computacionales recurriendo al ambiente desarrollado para tal fin. Los resultados de estas experiencias computacionales se detallan en [6]. En este artículo nos concentraremos en los resultados de las pruebas en el mar. La primera prueba fue desarrollada en Peterhead (Escocia), desde el 18 al 23 de Agosto de 2004. La segunda prueba fue desarrollada en Burray, Islas Orcadas (Escocia) desde el 13 al 21 de Septiembre de 2004.

2.4 PRUEBAS EN PETERHEAD

El propósito principal de estas pruebas fue probar el sistema de navegación (GPS), las estrategias del módulo ATModule, la conectividad entre AUTOTRACKER y el software/hardware nativo del AUV. Estas pruebas se desarrollaron en superficie, en una bahía tranquila. Se usó un simulador de MBE para simular un oleoducto, que era inexistente en dicha bahía, para efectuar una prueba del tipo "hardware-in-the-loop". Se corrieron trece misiones, y el objetivo principal se alcanzó en gran parte. La Figura 4 muestra los resultados de la misión 12 en la

pantalla del programa SeeTrack¹. La forma de ocho obedece a una maniobra de posicionamiento inicial en dirección y sentido de la tubería por parte del AUV, y a una búsqueda inicial para empezar la inspección. Posteriormente el AUV describe una trayectoria de búsqueda, en la que se puede observar el radio de giro del vehículo.

2.5 PRUEBAS EN ORKNEY ISLANDS

El objetivo de estas pruebas fue verificar el sistema de navegación inercial del AUV sumergido y los módulos de seguimiento (MBETracer y SFM), así como repetir pruebas, pero ahora con un oleoducto real y a mar abierto.

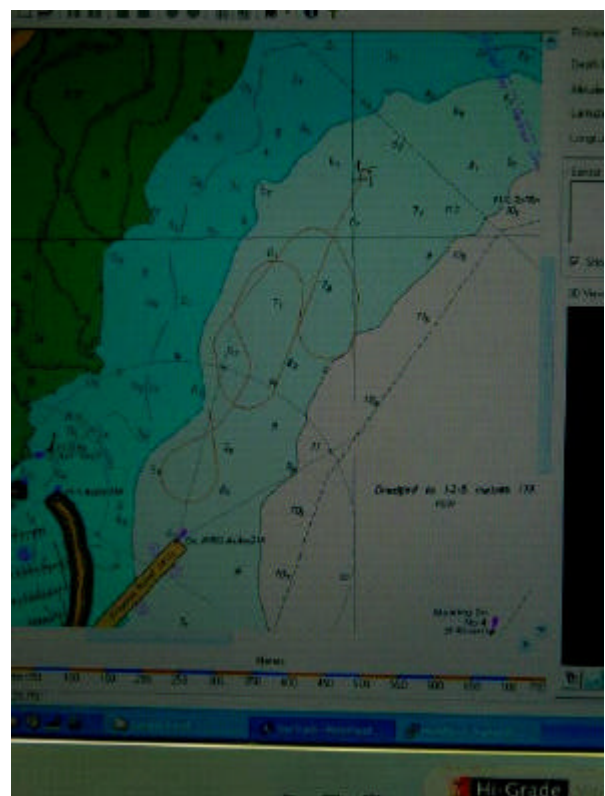


Figura 4: Misión 12 en Peterhead como se ve en la interfaz del software SeeTrack.

Ambas aproximaciones (ME y SE) se probaron en paralelo para comparar su consistencia. Como se resumen en la TABLA I, en la mayoría de los casos ambos sistemas concluyeron lo mismo. Esto se debe a que los experimentos desarrollados en esta primera etapa no fueron de la suficiente complejidad como para evidenciar un mejor comportamiento de EN4AUV como sería esperable. Esta conclusión guiará el diseño de futuras pruebas.

¹ cortesía de la empresa SeeByte Ltd., Scotland

Si bien estas fueron consideradas las pruebas finales del proyecto AUTOTRACKER, por falta de tiempo muchos postulados iniciales quedaron por validarse experimentalmente. Desde el punto de vista mecánico y del comportamiento del hardware, las pruebas a grandes profundidades. Desde el punto de vista algorítmico, la re-planificación, fue hecha fundamentalmente en 2D (a una altura de navegación fija), aunque está previsto en EN4AUV que los comportamientos en diferentes escenarios pudiesen involucrar un acercamiento o alejamiento del objetivo según el caso.

Tabla 1: Resumen de Misiones en Orkney Islands.

MISIÓN	OBJETIVO	STATUS	DESCRIPCIÓN
1 to 5	Prueba de los sistemas nativos del AUV y el hardware de detección y seguimiento del objetivo.	Éxito parcial	Problemas de comunicación entre la red interna del AUV y los paquetes de datos provenientes del sonar (MBE) no permitieron la adquisición. Sin embargo los paquetes de datos generados por sensores eran correctos.
6	Seguimiento-Búsqueda-Vuelta al Inicio-Oleoducto simulado. Consistencia entre Sistema Experto y Máquina de Estados.	Fallo	Un error de configuración en el SFM causó que ATModule abortara la misión.
7	Seguimiento-Búsqueda-Vuelta al Inicio-Seguimiento with EN4AUV. Oleoducto simulado. Consistencia entre Sistema Experto y Máquina de Estados.	Éxito	ATModule completó la misión. La trayectoria que describió el AUV fue casi la misma que la predicha por el simulador ATMEnvS.
8 to 13	Simple	Fallo	Algunos

	Seguimiento de un oleoducto real.		parámetros mal ajustados en el SFM hicieron que la comunicación entre SFM y MBETracer fallase y no se pudiese efectuar una correcta detección.
14	Simple Seguimiento de un oleoducto real.	Abortada	Un nuevo error de configuración causó la misión se abortara antes de empezar.
15	Simple Seguimiento de un oleoducto real. Consistencia entre Sistema Experto y Máquina de Estados.	Éxito	ATModule completó la misión. El oleoducto fue correctamente inspeccionado.

4 AUVI: EL PROTOTIPO DE BAJO COSTO

Con el propósito de contar con un prototipo de bajo costo, fundamentalmente de operación, que estuviese disponible todo el tiempo necesario en el grupo de I+D, se inició la construcción del prototipo de vehículo AUVI, financiado parcialmente por el proyecto con el mismo acrónimo y por la Universidad de las Islas Baleares. La idea central es poder continuar con la posibilidad de emplear el mismo ambiente de desarrollo que de run-time, ya generado en el proyecto anterior, y que sólo requiera una adaptación mínima desde el punto de vista de la arquitectura de software. El ambiente de simulación fue separado del resto del software para que se comunique a través de un puerto serie, empleando un adaptador del protocolo NMEA y del Ocean Shell, que es el framework de comunicaciones que se viene empleando. De este modo, el reemplazo del simulador por este prototipo puede realizarse en forma sencilla y directa. La arquitectura adaptada para este nuevo prototipo puede verse en la siguiente figura 5. En ella se conserva el mismo planificador dinámico de trayectorias (ATModule + OAS), y el mismo marco de comunicaciones por mensajes entre los módulos (OS). El sistema de guiado es por línea

de vista (LOS) y el de control, actualmente es un simple controlador proporcional que minimiza el ángulo $b = l - \text{rumbo del vehículo}$, empleando la de la ec. (1) y la información proveniente de sensores (referirse a la figura 6). Para el seguimiento del objetivo se emplea un sonar TriTech Seaking con su respectivo driver. Para la evasión de obstáculos, un sonar ST-30. En las primeras pruebas previstas para Julio de 2006 se realizará una navegación autónoma en 2D, para lo que se incorporaron un GPS (que proporciona la posición (x,y) y una brújula, que da la orientación o rumbo del vehículo (γ). El módulo

“observer” de la figura 5, simplemente “escucha” el broadcast de los mensajes de posición para armar una interfaz amigable con el usuario (HMI) que sirve para introducir las especificaciones estáticas de la misión además de para la visualización de la trayectoria que va describiendo el vehículo.

$$(1) \quad \mathbf{I} = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right)$$

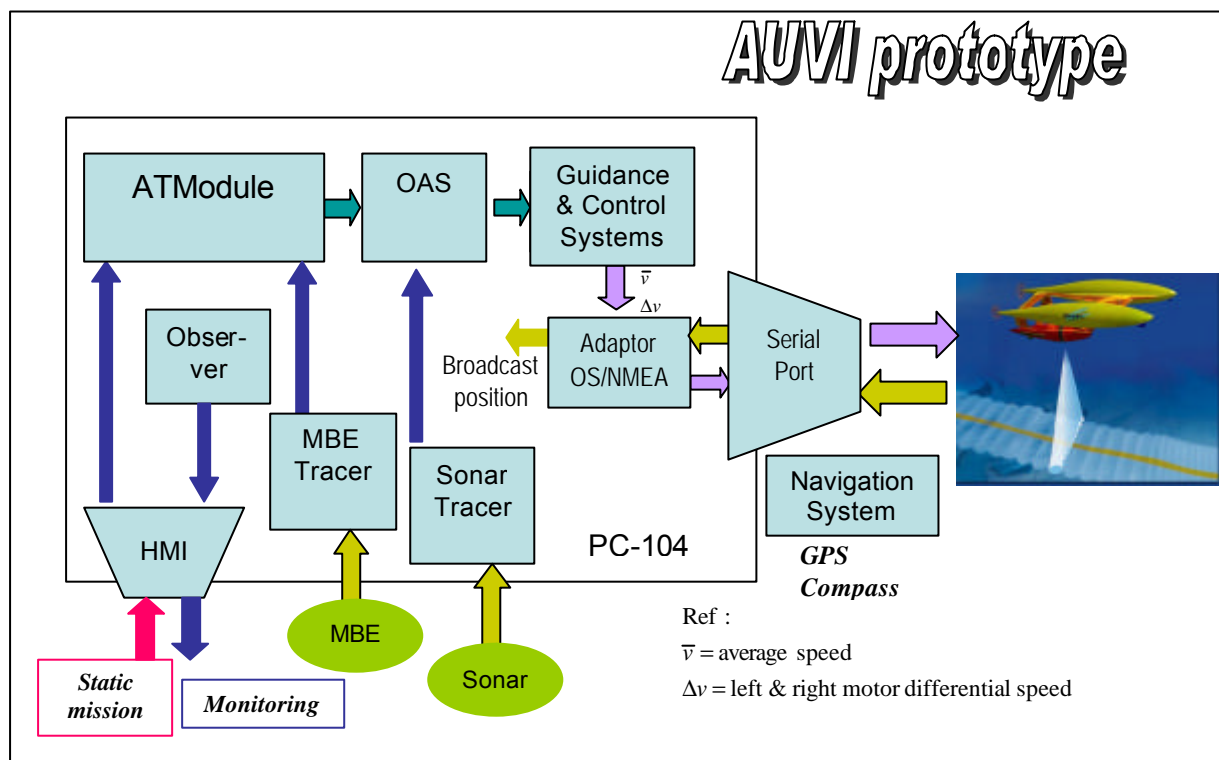


Figura 5: adaptación de la arquitectura de la figura 1 al nuevo prototipo AUVI.

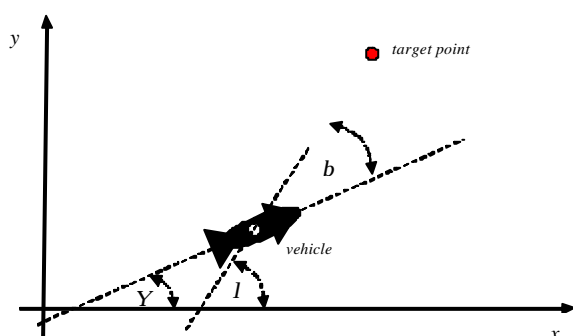


Figura 6: Guiado por línea de vista.

Desde un punto de vista mecánico, AUVI consta de dos cascos en forma de torpedo, unidos solidariamente conformando una estructura tipo catamarán, como puede apreciarse en la figura 7.



Figura 7: Prototipo de bajo costo AUVI.

5 CONCLUSIONES

Se han presentado en este artículo aspectos importantes sobre las consideraciones fundamentales en el diseño del módulo de software encargado de la re-planificación de la trayectoria deseada para un vehículo submarino autónomo, en función de la planificación de misión estática y de un entorno cambiante. Se describieron ambas aproximaciones construidas, máquina de estados y sistema experto, y se mostraron los resultados de las pruebas en el mar del Norte. Particularmente, la solución con inteligencia artificial demostró su viabilidad para el crecimiento incremental a medida que se adquiriera más conocimiento sobre inspecciones de oleoductos y otras aplicaciones, considerando el problema adicional del mantenimiento de consistencia en la base de conocimiento. Aunque aun quedan pruebas pendientes, estos experimentos realizados dentro del proyecto AUTOTRACKER sientan las bases para un prototipo listo para ser transferido, y por lo tanto, demuestran que la tecnología empleada está lo suficientemente madura como para enfrentar el desafío de la inspección autónoma submarina en aguas profundas. A partir de la experiencia ganada y de los temas de I+D abiertos que surgieron, se comenzó con la construcción de un prototipo de bajo costo, el vehículo AUVI, que sirva de banco experimental para continuar perfeccionando algoritmos y en particular, enriquecer la base de conocimiento actual del sistema experto. Esta construcción también se describió brevemente aquí.

Agradecimientos

Este trabajo fue financiado en el 5th European Community Framework Programme (AUTOTRACKER Project GDR1-2000-25150), y por la British Petroleum Co. (pruebas Peterhead y Orkney). También fue parcialmente financiado por Marie-Curie International Fellowship, 6th European Community Framework Programme (Project 03027 – AUVI).

Referencias

- [1] A. Balasuriya and T. Ura, "Autonomous Target Tracking by Underwater Robots Based on Vision", IEEE Proc. Int'l Symposium on Underwater Technology, 15-17 April 1998, Tokyo, Japan, pp. 191-197.
- [2] Antonelli, G.; Chiaverini, S.; Finotello, R.; Schiavon, R., "Real-time path planning and obstacle avoidance for RAIS: an autonomous underwater vehicle", IEEE Journal of Oceanic Engineering, Vol. 26, Is.2, April 2001, pp. 216 – 227.
- [3] J. Evans, Y. Petillot, P. Redmond, M. Wilson and D. Lane, "AUTOTRACKER: AUV Embedded Control Architecture for Autonomous Pipeline and Cable Tracking", Proceedings of IEEE-OCEANS 2003, Volume 5, 22-26 Sept. 2003, pp. 2651-2658.
- [4] I. Tena Ruiz, Y. Petillot, D.M. Lane, "Improved AUV navigation using side-scan sonar", Proceedings of IEEE OCEANS 2003, Volume: 3, 22-26 Sept. 2003, pp. 1261 – 1268.
- [5] Y. Petillot, I. Tena Ruiz, D.M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar", IEEE Journal of Oceanic Engineering, Volume: 26, Issue: 2, April 2001, pp. 240 – 251.
- [6] H. J. Curti, G. G. Acosta, O. A. Calvo "Autonomous Underwater Pipeline Inspection in AUTOTRACKER PROJECT: the Simulation Module", also in this Proceedings of IEEE-OCEANS 2005.
- [7] G. G. Acosta, H. J. Curti, O. A. Calvo, and J. Mochnacs, "An Expert Navigator for an Autonomous Underwater Vehicle", Proceedings of SADIO/Argentine Symposium on Artificial Intelligence ASAI '03, Buenos Aires, Argentina, Sept. 3-5 2003, in CD.

CREACIÓN DE LABORATORIOS VIRTUALES Y REMOTOS USANDO EASY JAVA SIMULATIONS Y LABVIEW. “EL SISTEMA HEATFLOW COMO UN CASO DE ESTUDIO”

H. Vargas, R. Dormido, N. Duro, S. Dormido-Canto
Dpto. de Informática y Automática, UNED
hvargas@bec.uned.es, {raquel, nduro, sebas}@dia.uned.es

Resumen

El presente artículo muestra el estado de desarrollo de un completo laboratorio para la realización de experiencias en automática mediante sesiones de experimentación virtual y remota a distancia: “El sistema HeatFlow”. El laboratorio permite a los usuarios trabajar tanto en modo virtual como remoto con la misma interfaz gráfica. El lado del servidor utiliza LabVIEW de National Instruments para la adquisición de datos, generación del lazo de control y comunicación a través de Internet. La interfaz del cliente ha sido creada utilizando el sistema de desarrollo conocido como Easy Java Simulations, una herramienta open source para generar potentes aplicaciones y applets sin la necesidad de tener conocimientos avanzados en programación.

Palabras Clave: Educación a distancia, laboratorio virtual y remoto, control automático.

1 INTRODUCCION

Los laboratorios basados en web son entornos de experimentación en el que los usuarios pueden operar sobre una serie de componentes gráficos, cada uno representando un elemento importante del experimento, y llevar a cabo la tele-operación o manipulación remota de un laboratorio virtual (donde la interfaz de experimentación trabaja contra una simulación del proceso) o remoto (donde el proceso es real y accesible a través de Internet) con el simple manejo del ratón o el teclado. De esta manera, los laboratorios basados en web extienden el área de aplicación de las tradicionales sesiones de los laboratorios presenciales con entornos de experimentación virtual y remota a distancia.

Muchas son las áreas del conocimiento e investigación en ingeniería que se han nutrido de los avances de las tecnologías basadas en Internet y la importancia que esta ha tenido en la enseñanza-aprendizaje basada en web. El control automático es

una de las áreas técnicas de la ingeniería en la que el impacto de estas tecnologías ha sido especialmente significativo en todo lo relativo al desarrollo de herramientas para el aprendizaje y la experimentación [1] y [5]. En un laboratorio de control basado en web, los estudiantes pueden acceder al laboratorio desde una localización remota que les permite interactuar con el experimento real. Es posible cambiar los parámetros de control, ejecutar los experimentos, descargar datos y visualizar resultados a través de Internet. Con esta perspectiva, el Departamento de Informática y Automática de la UNED en España está trabajando en el desarrollo de nuevos paradigmas de laboratorios para la realización a través de Internet de experiencias prácticas de ingeniería de control sobre plantas y sistemas reales o simulados.

En este contexto, el sistema *HeatFlow* presenta características muy interesantes para la investigación y la educación de los fundamentos de control. Es posible desarrollar un modelo sencillo de primer orden y su control asociado como primer acercamiento a las técnicas de identificación, para luego en subsiguientes experimentos incrementar el grado de dificultad analizando técnicas de identificación y control de sistemas con retardo de tiempo.

El artículo se organiza de la siguiente manera. El apartado 2 introduce los conceptos básicos de *Easy Java Simulations*. En la apartado 3 se describe el sistema de control de flujo de calor “*Heatflow*”. El apartado 4 presenta la construcción del laboratorio remoto. Cada sección de este apartado muestra: la confección de la vista del laboratorio elaborada en *Ejs* (el cliente) y la implementación del servidor. Esta última, contiene dos sub-secciones (4.2.1 y 4.2.2) que introducen aspectos relacionados con la ubicación del controlador en el laboratorio remoto y el diseño y construcción de la aplicación servidora en LabVIEW. El apartado 5 describe la lógica de control de acceso a los recursos del laboratorio virtual y remoto y para finalizar, el apartado 6 recoge algunas conclusiones y consideraciones generales del trabajo.

2 EASY JAVA SIMULATIONS

Ejs es una herramienta de software gratuita diseñada para la creación de simulaciones interactivas en Java [4]. El usuario al que está dirigida *Ejs* son estudiantes, profesores e investigadores de ciencias e ingeniería, que poseen un conocimiento básico de programación de computadores, pero que no disponen de una gran cantidad de tiempo para crear una simulación gráfica con un elevado grado de interactividad.

Las simulaciones en *Ejs* son estructuradas en dos partes, el *Modelo* y la *Vista*. En el modelo se describe el comportamiento del sistema mediante variables y código Java o ecuaciones diferenciales ordinarias. Mientras que la vista provee el aspecto gráfico o visual de la simulación. La interfaz de usuario de *Ejs* se presenta en la figura 1.

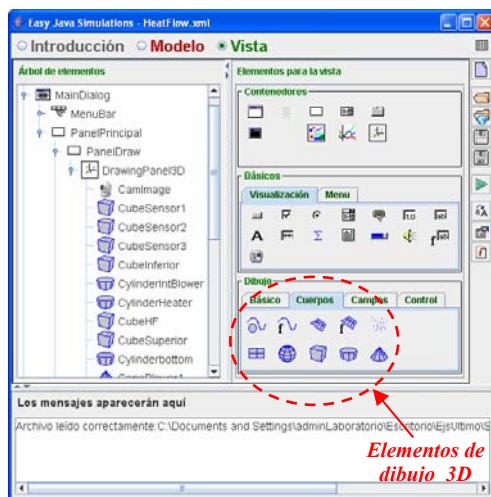


Figura 1: Interfaz gráfica de usuario de *Ejs* para la creación de la vista de una simulación. El árbol de la izquierda corresponde a la vista de la figura 4.

Ejs se puede utilizar de forma independiente para crear simulaciones interactivas, ya sea como aplicación o como applet, sin embargo, también es posible utilizar *Ejs* para agregar interactividad a modelos Simulink [2].

3 EL SISTEMA HEATFLOW

Para el desarrollo del laboratorio virtual y remoto del trabajo se ha utilizado el sistema Heatflow fabricado por la empresa canadiense Quanser Consulting [12].

La planta consiste en una caja equipada con los siguientes componentes: un calefactor y un ventilador ubicados en un extremo de la estructura y tres sensores de temperatura S1, S2 y S3 localizados

en diferentes posiciones. En la figura 2 se muestra la estructura completa de la planta. La potencia liberada al calefactor es controlada mediante una señal analógica. Igualmente, la velocidad del ventilador puede ser controlada usando una señal analógica. Para obtener la temperatura se usan transductores de platino por su rapidez de asentamiento en la medición. La velocidad de giro del ventilador se mide utilizando un tacómetro y se puede usar para diseñar controladores de velocidad.

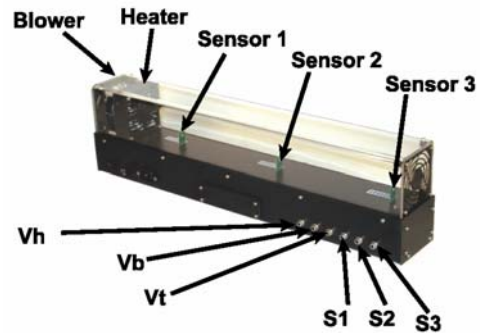


Figura 2: El sistema HeatFlow de Quanser.

En principio, puede definirse un modelo genérico de la forma:

$$\dot{T}_n = F(V_h, V_b, T_a, x_n) \quad (1)$$

Donde:

- T_n es la temperatura en el sensor n .
- V_b es el voltaje aplicado al ventilador.
- V_h es el voltaje aplicado al calefactor.
- T_a es la temperatura ambiente.
- x_n es la distancia del sensor n al calefactor.

Debido a que un modelo matemático completo que describa la termodinámica del sistema es difícil de derivar, se han estimado los parámetros del modelo mediante técnicas de identificación. Para ello, se procede a determinar las funciones de transferencia para cada sensor de temperatura en función del voltaje aplicado al calefactor. A partir de experiencias mediante respuesta a escalón en lazo abierto se registran las respuestas del proceso para cada sensor. Para que el registro sea válido, la salida debe haber alcanzado el estacionario. En cada experiencia, el voltaje aplicado al ventilador se mantiene constante y se utiliza como variable asociada a una *perturbación* para las funciones de transferencia resultantes del sistema. La identificación final de los modelos se ha realizado

con el toolbox de identificación de sistemas de Matlab - IDENT.

Bajo estas consideraciones, el modelo que mejor se ajusta al comportamiento del sistema es de la forma:

$$G(s) = \frac{T_n(s)}{V_h(s)} = \frac{K_p (1 + \tau_3 s) e^{-\tau_d s}}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (2)$$

- $T_n(s)$ es la temperatura en el sensor n .
- $V_h(s)$ es el voltaje aplicado al calefactor.

donde la ganancia, constantes de tiempo y el retardo dependen de cual de los tres sensores se está utilizando para cerrar el lazo de control de temperatura.

4 LABORATORIO VIRTUAL Y REMOTO

La arquitectura del laboratorio remoto se basa en la conocida estructura cliente/servidor [3]. El ordenador que implementa el lado del servidor contiene el servidor web que da acceso al applet y el sistema de control en tiempo real (figura 3). La interfaz del cliente se describirá en la sección 4.1, y corresponde a la vista construida con *Ejs*. Esta vista interactiva puede funcionar como laboratorio virtual utilizando el modelo matemático de la planta o como laboratorio remoto.

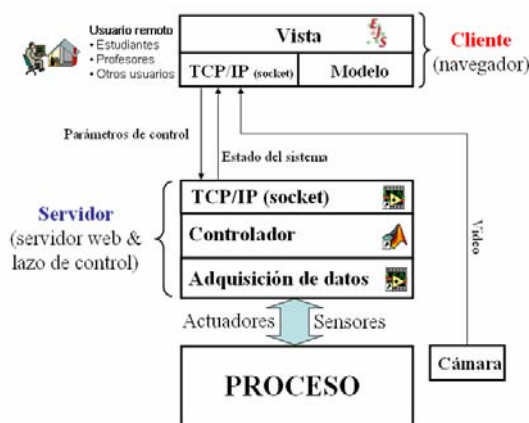


Figura 3: Arquitectura cliente - servidor.

La implementación del servidor se describirá en detalle en la sección 4.2. Desde LabVIEW se realizan las tareas de control de acceso, transmisión continua de datos entre el cliente y el servidor, comunicación con el sistema físico a través de tarjetas de adquisición de datos y cierre del lazo de control.

4.1 EL CLIENTE

En la figura 4 se muestra la ventana principal del laboratorio virtual y remoto. La parte superior contiene una representación 3D del sistema heatflow cuyo color varía en función del estado del proceso (se utiliza el modelo identificado en el apartado anterior cuando el sistema se ejecuta como laboratorio virtual; cuando se utiliza la opción remota el sistema accede al proceso real del laboratorio).

La parte inferior de la ventana principal contiene un panel de control que permite escoger el tipo de experiencia (virtual o remota), y un conjunto de sliders y botones que permiten definir diversas situaciones en la dinámica del proceso (por ejemplo, realizar un cambio en la consigna o introducir alguna perturbación variando el voltaje en el ventilador).

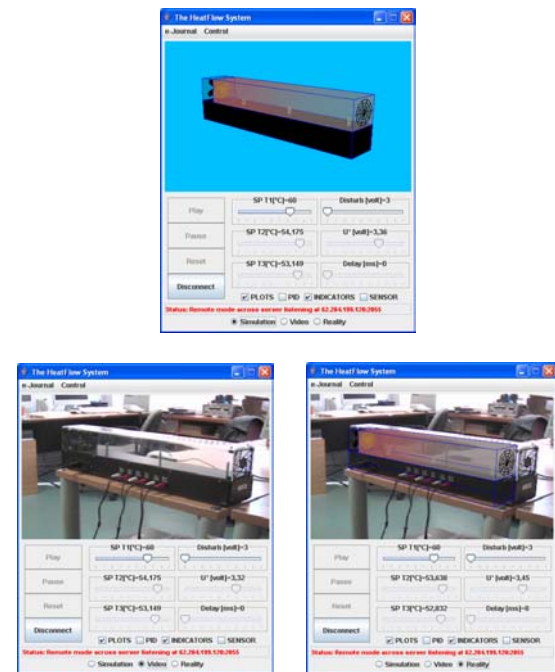


Figura 4: Interfaz gráfica de usuario (vista *Ejs*) creada con la librería de elementos gráficos de *Ejs*.

Los cuatro botones localizados a la izquierda del panel de control (*Play*, *Pause*, *Reset*, y *Remote*) permite al usuario controlar las operaciones principales sobre la evolución del sistema. Si se pulsa el botón *Remote* el sistema se conectará a la planta real ubicada en el laboratorio de la universidad, si el nombre de usuario y la contraseña introducida son correctas (esta característica se describe en la sección 4.2). Si este botón no se acciona la aplicación trabaja en modo simulación. A la derecha de los botones, seis sliders permiten ajustar los valores de las consignas para cada sensor de temperatura (*SP T1*, *SP T2*, y *SP T3*), el voltaje aplicado al ventilador (*Disturb*), el voltaje suministrado al calefactor (*U**),

y ajustar un retardo en la simulación (*Delay*). Adicionalmente, cuatro botones de selección permiten la apertura de cuatro ventanas auxiliares. Si se selecciona la opción *PLOT* se visualiza un panel compuesto por dos gráficas en las que se recoge la evolución de las variables más importantes del proceso, es decir, las medidas de los sensores de temperatura y la acción de control U^* sobre el calefactor; seleccionando la opción *PID* se obtiene un panel con los parámetros del controlador PID asociado a cada sensor. El botón *INDICATORS* despliega una nueva ventana con un conjunto de campos numéricos que muestran el valor de las distintas variables de interés del proceso. El botón *SENSOR* permite escoger el sensor para cerrar el lazo de control de temperatura.

Un campo de texto no editable ubicado debajo del panel de control sirve para visualizar mensajes sobre el estado de la conexión entre el cliente (vista *Ejs*) y el servidor (planta real).

Tres botones radiales (*Simulation*, *Video*, y *Reality*) localizados en la parte más baja de la aplicación permiten seleccionar tres formas diferentes de ver la planta (figura 4). El primero muestra la planta creada con los elementos propios de *Ejs*; el segundo, muestra el proceso real que está en el laboratorio remoto; y el tercero, permite ver la planta real en el laboratorio remoto y superpuesto a ella la planta simulada.

Un menú desplegable (*e-Journal*) localizado en la parte más alta de la consola muestra un sub-menú con cinco botones: *SaveGraph*, *SaveData*, *ReadData*, *Start Record*, y *Stop Record* que permiten guardar una imagen de la evolución de las variables en formato .gif, salvar el estado del sistema en un fichero de datos binario .dat, leer un fichero de datos binario .dat, grabar en un fichero .m los parámetros de los controladores y los valores de la evolución temporal de las variables controladas y manipuladas, y parar la grabación de datos (esta opción está activa una vez que la grabación ha comenzado). Finalmente, otro menú desplegable (*Control*) localizado a la derecha del anterior permite conmutar las acciones de control de modo manual a automático, y viceversa.

4.2 EL SERVIDOR

La descripción de la aplicación servidora está dividida en dos sub-secciones: Ubicación del lazo de control y la implementación en LabVIEW. La primera de ellas, describe la alternativa elegida para la localización del lazo de control, mientras la segunda presenta la estructura y descripción funcional del diseño, análisis y construcción de la aplicación que implementa el lado del servidor en LabVIEW.

4.2.1 Ubicación del lazo de control

Existen dos políticas posibles de considerar respecto de la ubicación del controlador en el sistema. La primera de ellas considera el *cierre del lazo de control a través de la interfaz remota* y la segunda apunta a ubicar el *control en el mismo ordenador que se encuentra conectado a la planta*.

Debido a que el cierre del lazo de control a través de la interfaz remota presenta serios inconvenientes provocados por los retardos variables que se producen en las transmisiones, más aun, pensando que se pretende utilizar la red Internet tradicional para el acceso al laboratorio remoto por parte de los usuarios. Estos retardos pueden provocar inesperadas inestabilidades en la planta, que se incrementan cuando la dinámica de la planta es rápida.

Por estas razones, para el prototipo del sistema *heatflow* se ha escogido la segunda opción, tal y como se muestra en la figura 5. De esta forma, el controlador se sitúa en el ordenador remoto directamente conectado a la planta y la interfaz del cliente se limita a realizar labores de supervisión.

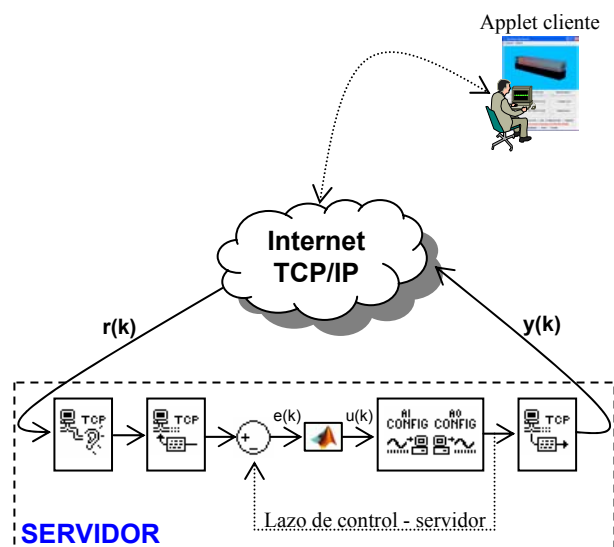


Figura 5: Cierre del lazo de control en el servidor.

En este sentido, mediante los controles del entorno de experimentación el usuario puede observar los valores de determinadas señales, modificar variables e incluso introducir perturbaciones, sin embargo, todo el proceso de cálculo junto a la transmisión y recepción de datos entre el controlador y la planta real se realiza siempre en forma local en el servidor remoto.

Más detalles respecto de este punto crítico en el desarrollo de un laboratorio remoto puede encontrarse en [9].

4.2.2 Implementación en LabVIEW

La interfaz del cliente ya se ha descrito en la sección 4.1, ya que se trata de la vista construida con *Ejs*. Esta vista interactiva puede funcionar como laboratorio virtual utilizando un modelo matemático de la planta o como laboratorio remoto. En este caso, la vista establece una conexión TCP con el controlador del lado servidor. Esta conmutación se produce cuando el usuario pulsa el botón *remote* de la vista. Desde ese momento, la aplicación cliente envía un array de datos al servidor, obteniendo como respuesta un vector con el estado actual de la planta. De esta manera, el lazo de control opera sobre la planta en función de los paquetes de datos enviados desde la vista.

Los parámetros de control remitidos desde la vista al controlador son: modo de control (Manual/Automático), sensor (S1, S2, S3), voltaje manual al calefactor (U^*), parámetros de los controladores PID ($Kp1$, $Ti1$, $Td1$, $Kp2$, $Ti2$, $Td2$, $Kp3$, $Ti3$, $Td3$), consignas de temperatura ($sp1$, $sp2$, $sp3$) y voltaje manual al ventilador (Disturb), formando un paquete de datos de 60 *bytes*. La información que la vista obtiene como respuesta está compuesta de cinco valores: instante de muestreo (t), temperatura en los sensores (S1, S2, S3) y voltaje aplicado al calefactor (U^*). El tamaño de este paquete de datos es de 20 *bytes*.

El lazo de control en el servidor se ha desarrollado utilizando *LabVIEW* [12] y *Matlab/Simulink* [14]. El intercambio de datos entre la planta y el ordenador se lleva a cabo mediante una tarjeta de adquisición de datos DAQ de National Instruments desde *LabVIEW*, donde además se cierra el lazo de control y se configura el socket servidor que queda a la espera de conexión desde el cliente. Sin embargo, los controladores PID están programados en *Matlab/Simulink*, y se accede a ellos mediante un bloque para llamar y evaluar un *scripts* de Matlab desde programas *LabVIEW*. Esta estructura de diseño del servidor presenta una gran flexibilidad para incluir nuevas estrategias de control (por ejemplo, controladores predictivos o difusos) con un pequeño esfuerzo en programación.

Adicionalmente, una base de datos *MySQL* en el servidor se encarga de la gestión de usuarios y reserva de recursos de acceso a la planta real localizada en el laboratorio de la universidad (esta característica se describe en el apartado 5). Cada registro en la base de datos corresponde a una reserva del recurso realizada por un usuario para una fecha y rango de tiempo establecido y que se deberá verificar cada vez que un cliente desee conectarse al recurso remoto ubicado en el laboratorio de la universidad.

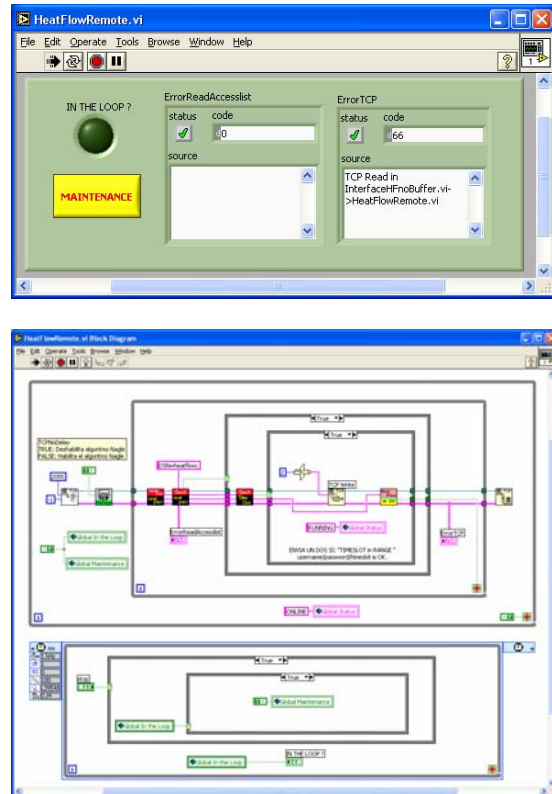


Figura 6: Panel frontal y diagrama de bloques de la aplicación servidora en LabVIEW.

La figura 6 muestra el panel frontal y diagrama de bloques de la aplicación servidora programada en *LabVIEW*. El panel frontal es muy simple e intenta disminuir la cantidad de elementos gráficos con el objetivo de optimizar el rendimiento de la aplicación. Sólo cuenta con dos indicadores que permiten visualizar un posible error de conexión con la base de datos o de comunicación con la aplicación cliente, un led denominado *IN THE LOOP* que indica si un usuario se encuentra o no operando la planta remota y por último un botón denominado *MAINTENANCE* que permite parar la aplicación. El diagrama de bloques representa la programación por flujo de datos que define la funcionalidad de la aplicación *LabVIEW*.

La tabla 1 presenta un resumen descriptivo de los principales bloques utilizados. Internamente, cada objeto contiene un nuevo diagrama de bloques que implementa la funcionalidad que se describe a la derecha de cada VI de la tabla.

Cuando un cliente intenta conectar con el servidor para realizar una sesión experimental a distancia es detectado por el bloque *TCP Listen*. Una vez que la conexión se establece, el bloque *TCP No Delay* deshabilita el algoritmo Nagle (envío de un acuse de recibo por cada segmento de datos transmitido) de la conexión que viene habilitado por defecto [9].

Tabla 1: Bloques (VIs) de la aplicación servidora.

VI	Descripción
	<i>TCP Listen</i> : Crea un socket de escucha y espera por una conexión de red TCP desde el cliente (nativo de LabVIEW).
	<i>TCPNoDelay</i> : Deshabilita el algoritmo Nagle de una conexión TCP/IP establecida entre el cliente y el servidor.
	<i>TCP Write</i> : Escribe datos sobre una conexión o canal de red TCP (nativo de LabVIEW).
	<i>TCP Read</i> : Lee un número específico de bytes desde una conexión de red TCP (nativo de LabVIEW).
	<i>Read User/Pass</i> : Lee el usuario y password (encriptada en MD5) del cliente desde la conexión TCP.
	<i>Check User/Pass</i> : Consulta la base de datos y verifica si el usuario y password existe.
	<i>Check Time Slot</i> : Si existe el usuario y password anterior, verifica si la fecha y hora de reserva son correctas.
	<i>Acquisition Data/Control</i> : Acceso a la planta real y cierre del lazo de control. Realiza el intercambio continuo de datos entre el cliente y el servidor.
	<i>TCP Close</i> : Cierra una conexión TCP establecida entre la aplicación cliente y el servidor (nativo de LabVIEW).

El bloque *Read User/Pass* (internamente contiene el bloque *TCP Read*) obtiene el nombre de usuario y contraseña despachada por la aplicación Ejs cliente y los entrega al bloque *Check User/Pass* para consultar la tabla *AccessList* de la base de datos que contiene las reservas horarias del laboratorio y verificar si el nombre de usuario y contraseña existe. Si el nombre de usuario y contraseña existen, entrega el campo *starttime* (fecha y hora de inicio de la reserva) y *endtime* (fecha y hora de fin de la reserva) del registro encontrado al bloque *Check TimeSlot* que se encarga de verificar si la fecha y hora de reserva son las correctas. Si todas las verificaciones anteriores son positivas, entonces se habilitará el acceso del usuario a la planta remota mediante el bloque *Acquisition Data/Control* que realiza el intercambio sostenido de datos entre el cliente y el servidor durante la sesión experimental.

En cada iteración del bucle interno de este bloque se realiza la lectura de la estructura de control enviada desde el cliente al servidor, la escritura de la trama de estado desde el servidor al cliente y la comparación del tiempo actual con el tiempo de fin de la sesión de experimentación con el objetivo de cerrar la sesión cuando este tiempo final se haya sobrepasado. Ante

cualquier evento de término de la conexión (fin de la sesión experimental o corte inesperado de la conexión a Internet) se anula la acción sobre los actuadores del sistema enviando un valor de cero voltios y, finalmente, se cierra la conexión con el cliente utilizando el bloque *TCP Close*.

5 ACCESO A LOS RECURSOS

El acceso final a los recursos del laboratorio por parte de los usuarios se lleva a cabo a través del portal de CYBERLAB. Cyberlab define una arquitectura de laboratorios orientada a proporcionar un servicio independiente de gestión de usuarios y reserva de uso de los laboratorios adscritos a la red Cyberlab, para facilitar la integración de laboratorios remotos y virtuales de diferentes instituciones (figura 7).



Figura 7: Sistema de reservas de CYBERLAB.

Cyberlab define dos tipos de usuarios distintos: proveedores y clientes. Los proveedores se encargan de proporcionar el laboratorio remoto e implementar el mecanismo definido por Cyberlab para indicar la disponibilidad de los experimentos y asignación de tiempos que un usuario final del laboratorio puede realizar. Ese mecanismo lo constituye un servidor central (Cyberlab Provider) que informa al proveedor del laboratorio sobre las conexiones de los usuarios y las reservas de espacios de tiempo generadas a través del portal Cyberlab actualizando la base de datos del proveedor con estas nuevas reservas. Los clientes son aquellos usuarios del portal Cyberlab que realizan reservas de tiempo sobre los laboratorios y acceden a ellos en dichos espacios temporales. El acceso final al laboratorio se hace a través de unas páginas HTML especialmente diseñadas para mostrar el applet de estado que permite al usuario identificarse y acceder al laboratorio.

Más detalles acerca del proveedor del servicio de reserva de recursos y su descripción funcional se pueden encontrar en [11].

6 CONCLUSIONES

La investigación en el campo de la experimentación virtual y remota en la educación de las ingenierías puede considerarse ya una tecnología madura. Sin embargo, aún es difícil encontrar en revistas especializadas en la educación del control la aplicación de estos elementos de aprendizaje a niveles de técnico de automatización. Lo anterior implica ignorar la existencia de este amplio y significativo grupo en el ámbito de la ingeniería del control. En este sentido, y con el objetivo de dar solución a estos problemas, cinco universidades Europeas están llevando a cabo un proyecto piloto denominado “AutoTECH: Automation Technicians Vocational Training Repository” [10], cuya meta principal implica desarrollar un conjunto de recursos de aprendizaje basado en web para satisfacer los requerimientos de estas personas. El sistema *heatflow* es uno de estos esfuerzos realizado por uno de los miembros del consorcio AutoTech, el grupo de investigación de la UNED.

En relación con el diseño, construcción y desarrollo del laboratorio virtual y remoto del artículo, *Ejs* ha demostrado ser una simple y poderosa herramienta para la construcción de sofisticados laboratorios virtuales y remotos en el campo de la automática. Las posibilidades de *Ejs* aún pueden ser aumentadas de forma relativamente sencilla mediante conexiones programadas desde el lado servidor de forma más intuitiva especialmente si se elige NI LabVIEW como herramienta para crear el lado servidor.

Agradecimientos

Los autores quieren agradecer la financiación de este trabajo a la CICYT en el marco del proyecto DPI2004-01804 y a la Comisión Europea en el marco del proyecto "Leonardo da Vinci Pilot- Project. N° 2004 N/04/B/PP 165.011".

Referencias

- [1] Dormido, S. (2004) “Control learning: Present and future” Annual Control Reviews, 28, pp. 115-136.
- [2] Dormido, S., Esquembre, F., Farías, G., Sánchez, J. (2005). Adding interactivity to existing Simulink models using Easy Java Simulations, 44th IEEE CDC/ECC, Sevilla (Spain).
- [3] Duro, N., Dormido, R., Vargas, H., Dormido, S., Sánchez, J., Pastor, R. (2005). The Three Tank System: A Remote and Virtual Control Laboratory using Easy Java Simulations, 44th IEEE CDC/ECC, Sevilla (Spain).
- [4] Esquembre, F. (2004). Easy Java Simulations: A software tool to create scientific simulations in Java, *Comp. Phys. Comm.*, 156 (2), pp. 199-204.
- [5] Guillet, D., Nguyen, A., Rekik, Y. (2005). Collaborative Web-Based Experimentation in Flexible Engineering Education, *IEEE Trans on Education*, vol. 48, n° 4.
- [6] Pastor, R., Sánchez, J., Dormido, S. (2005). Web-Based Virtual Lab and Remote Experimentation Using Easy Java Simulations, 16th IFAC World Congress, Prague (Czech Republic).
- [7] Valera, A., Diez, J.L., Vallés, M., Albertos, P. (2005). Virtual and Remote Control Laboratory Development, *IEEE Control System Magazine*, 25 (1), pp. 35-39.
- [8] Vargas, H., Farías, G., Dormido, S., Sánchez, J. (2006). Web-Based Learning Resources for Automation Technicians Vocational Training: Illustrated with a Heatflow and Liquid Level Laboratories, 7th ACE-IFAC, Madrid (Spain).
- [9] Sánchez, J. (2001). “Un nuevo enfoque metodológico para la enseñanza a distancia de asignaturas experimentales: Análisis, diseño y desarrollo de un laboratorio virtual y remoto para el estudio de la automática a través de Internet”. Tesis doctoral, UNED, Madrid.
- [10] AutoTech (2006). <http://www.pidstop.com>
- [11] Cyberlab (2006). <http://www.cyberlab.org>
- [12] LabVIEW (2006). <http://www.ni.com/labview>
- [13] Quanser (2006). <http://www.quanser.com>
- [14] Matlab (2006). <http://www.mathworks.com>

UNA DEMOSTRACIÓN DE LA PRUEBA DE B.I.B.O. ESTABILIDAD PARA SISTEMAS DISCRETOS

Ekaitz Zulueta, Teodoro Rico, Idoia Manero, M. Karmele L. De Ipiña

E.U.I. de Vitoria-Gasteiz (EHU-UPV), Nieves Cano 12 C.P.01006

ekaitz.zulueta@ehu.es, teodoro.rico@ehu.es, idoia.manero@ehu.es, karmele.ipina@ehu.es

Pedro M. Iriondo

E.T.S.I.I. e I.T. de Bilbao, Alameda Urquijo s/n C.P.48013

Pedro.iriondo@ehu.es

Resumen

Se trata de demostrar el test de Jury. Con este test se puede comprobar que todas las raíces de un polinomio $P(z)$ de orden n ésimo, cumplen la condición de que su módulo es menor que la unidad, cumpliendo la condición de B.I.B.O de estabilidad del sistema.

Palabras Clave: Demostración del test Jury, Estabilidad B.I.B.O., dinámica de sistemas lineales.

1 INTRODUCCIÓN

Este trabajo es una demostración sencilla del test de Jury, que se basa en el trabajo de [7] sobre el test de Routh-Hurwitz. La demostración desarrollada en este trabajo permite a los alumnos de Automática comprender de una forma sencilla los pasos que se realizan en la prueba de Jury. Esto permitirá que los alumnos no se dediquen a memorizar la prueba sino que la comprendan hasta sus últimas consecuencias, puesto que resulta mucho más duradero en la mente de los ingenieros recordar aquello que entendieron que aquello que memorizaron. La demostración del test de Jury se puede abordar mediante el test de Routh-Hurwitz y aplicar la transformada bilineal. En cuyo caso la demostración no sería más el estudio de la demostración de Routh-Hurwitz. A pesar de que este procedimiento suena sencillo resulta complicado demostrar la relación de los coeficientes de un polinomio con los coeficientes de ese mismo polinomio después de haberle sido aplicado la transformación bilineal. En este trabajo no vamos a seguir ese método. Por otra parte, para la demostración del test de Routh-Hurwitz solamente se tiene acercar a estas referencias [1], [2], [3], [4] [5] y [6]. En estas referencias podrá encontrar los conocimientos previos necesarios para realizar la demostración del test de Routh-Hurwitz como son los índices de Cauchy y las cadenas de Sturms. En estas referencias también se explican la demostración

de la prueba de Routh-Hurwitz en sí misma, así como el estudio de los casos extremos en los cuales aparecen divisiones por cero. Pero en nuestra opinión resulta mucho más interesante para el estudiante de ingeniería la demostración del test de Routh-Hurwitz dada en [7]. De hecho, como se ha mencionado anteriormente esta demostración sigue un procedimiento parecido para la demostración del test de Jury.

2 PRUEBA O TEST DE JURY

Siendo:

$$P(z) = a_0 z^N + a_1 z^{N-1} + a_2 z^{N-2} + \dots + a_{N-1} z + a_N \quad (1)$$

y utilizando los coeficientes:

$$a_0 \quad a_1 \quad a_2 \quad \dots \quad a_{N-1} \quad a_N \quad (2)$$

$$a_N \quad a_{N-1} \quad a_{N-2} \dots a_1 \quad a_0 \quad (3)$$

$$\text{con } \eta = \frac{a_N}{a_0} \quad (4)$$

Restando los coeficientes, uno a uno y haciendo el término independiente 0 obtendremos

$$a'_0 \quad a'_1 \quad a'_2 \quad \dots \quad a'_{N-1} + 0 \quad (5)$$

$$a'_{N-1} \quad a'_{N-2} \quad a'_{N-3} \dots a'_0 \quad (6)$$

$$\text{con } \eta = \frac{a'_{N-1}}{a'_0} \quad (7)$$

De igual forma, obtendremos:

$$a''_0 \quad a''_1 \quad a''_2 \dots \quad (8)$$

Las raíces del polinomio $P(z)$ tendrán un módulo menor que 1 si:

$$a_0 \quad a'_0 \quad a''_0 \quad \dots > 0 \quad (9)$$

2.1 PRELIMINARES

Tomamos el polinomio $P(z)$ y calcularemos su polinomio recíproco $P_t(z)$, tal que:

$$P_t(z) = z^N P(z^{-1}) \quad (10)$$

Por tanto el polinomio $P_t(z)$ queda de la siguiente forma

$$P_t(z) = a_0 + a_1 z + a_2 z^2 + \dots + a_N z^N \quad (11)$$

Se define el polinomio siguiente:

$$Q(z) = P(z) - \eta P_t(z) \quad (12)$$

$$\begin{aligned} Q(z) &= a_0 z^N + a_1 z^{N-1} + \dots \\ &\dots + a_{N-1} z + a_N \\ &- \eta (a_N z^N + a_{N-1} z^{N-1} + \dots + a_1 z + a_0) \end{aligned} \quad (13)$$

$$\begin{aligned} Q(z) &= (a_0 - \eta a_N) z^N + \dots \\ &\dots + (a_{N-1} - \eta a_1) z + (a_N - \eta a_0) \end{aligned} \quad (14)$$

2.1.1 Teorema 1

Las raíces del polinomio $P(z)$ de modulo igual a la unidad serán raíces también de $Q(z)$ para cualquier valor real de η . También se puede afirmar que las raíces del polinomio $Q(z)$ que tengan módulo igual a la unidad, serán también raíces del polinomio $P(z)$, salvo para los valores de η de ± 1 .

Demostración:

Empezaremos demostrando que toda raíz del polinomio $P(z)$ de módulo uno también lo será de $Q(z)$. Si se supone de $P(z)$ tiene una raíz en $z = e^{j\omega}$ también será raíz de $P(z)$ $z = e^{-j\omega}$ puesto que los coeficientes del polinomio son reales.

$$P(z = e^{j\omega}) = 0 \Rightarrow P(z = e^{-j\omega}) = 0 \quad (15)$$

Debido a que $Q(z)$ se define de la siguiente forma, se puede demostrar fácilmente que cualquier raíz de módulo unidad del polinomio de $P(z)$ también lo será de $Q(z)$.

$$Q(z) = P(z) - \eta P_t(z) \quad (16)$$

$$\begin{aligned} Q(z) &= (z - e^{j\omega})^q (z - e^{-j\omega})^q P'(z) \\ &- \eta z^N \left(\frac{1}{z} - e^{j\omega} \right)^q \left(\frac{1}{z} - e^{-j\omega} \right)^q P' \left(\frac{1}{z} \right) \end{aligned} \quad (17)$$

Simplificando la ecuación anterior, se obtiene lo siguiente.

$$\begin{aligned} Q(z) &= (z - e^{j\omega})^q (z - e^{-j\omega})^q P'(z) \\ &- \eta z^{N-2q} (1 - ze^{j\omega})^q (1 - ze^{-j\omega})^q P' \left(\frac{1}{z} \right) \end{aligned} \quad (18)$$

$$\begin{aligned} Q(z) &= (z - e^{j\omega})^q (z - e^{-j\omega})^q P'(z) \\ &- \eta z^{N-2q} (z - e^{j\omega})^q (z - e^{-j\omega})^q P' \left(\frac{1}{z} \right) \end{aligned} \quad (19)$$

$$\begin{aligned} Q(z) &= (z - e^{j\omega})^q (z - e^{-j\omega})^q \\ &\left(P'(z) - \eta z^{N-2q} P' \left(\frac{1}{z} \right) \right) \end{aligned} \quad (20)$$

Como se puede observar que toda raíz de $P(z)$ de módulo la unidad y de índice de multiplicidad igual a q será raíz también de $Q(z)$ manteniendo su índice de multiplicidad. Como se puede observar estas raíces no varían con respecto al parámetro η .

Ahora se va a demostrar que todas las raíces de módulo unidad del polinomio $Q(z)$ son los mismos que tiene $P(z)$ para cualquier valor real de η excepto para ± 1 . Supongamos que para un valor concreto de η diferente de ± 1 , $Q(z)$ tiene una raíz de módulo 1 de índice de multiplicidad " q ".

$$Q(z = e^{j\omega}) = 0 \quad (21)$$

Ahora descompondremos el polinomio $P(z)$ en parte real y parte imaginaria, dejándolo en función de w .

$$P(z = e^{j\omega}) = \text{Re}(w) + j \text{Im}(w) \quad (22)$$

Si sustituimos en $Q(z)$ z por $e^{j\omega}$ podemos advertir que se obtiene la siguiente ecuación si se supone que $e^{j\omega}$ es raíz de $Q(z)$.

$$\begin{aligned} 0 &= \operatorname{Re}(\omega) + j \operatorname{Im}(\omega) \\ -\eta e^{j\omega N} [\operatorname{Re}(\omega) - j \operatorname{Im}(\omega)] \end{aligned} \quad (23)$$

$$\begin{aligned} 0 + j0 &= \operatorname{Re}(\omega) + j \operatorname{Im}(\omega) \\ -\eta (\cos(\omega N) + j \sin(\omega N)) \\ (\operatorname{Re}(\omega) - j \operatorname{Im}(\omega)) \end{aligned} \quad (24)$$

Igualando la parte real y la parte imaginaria del lado derecho de la ecuación anterior a cero, se obtienen las siguientes igualdades.

$$\begin{aligned} 0 &= \operatorname{Re}(\omega) - \eta \cos(\omega N) \operatorname{Re}(\omega) \\ -\eta \sin(\omega N) \operatorname{Im}(\omega) \end{aligned} \quad (25)$$

$$\begin{aligned} 0 &= \operatorname{Im}(\omega) - \eta \sin(\omega N) \operatorname{Re}(\omega) \\ +\eta \cos(\omega N) \operatorname{Im}(\omega) \end{aligned} \quad (26)$$

Simplificando las expresiones anteriores se llega al sistema de ecuaciones siguiente.

$$\begin{aligned} 0 &= (1 - \eta \cos(\omega N)) \operatorname{Re}(\omega) \\ -\eta \sin(\omega N) \operatorname{Im}(\omega) \end{aligned} \quad (27)$$

$$\begin{aligned} 0 &= -\eta \sin(\omega N) \operatorname{Re}(\omega) \\ + (1 + \eta \cos(\omega N)) \operatorname{Im}(\omega) \end{aligned} \quad (28)$$

Este sistema de ecuaciones se escribe de forma matricial así.

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 - \eta \cos(\omega N) & -\eta \sin(\omega N) \\ -\eta \sin(\omega N) & 1 + \eta \cos(\omega N) \end{bmatrix} \vec{v} \quad (29)$$

Donde \vec{v} se define así:

$$\vec{v}' = [\operatorname{Re}(\omega) \quad \operatorname{Im}(\omega)] \quad (30)$$

Como se puede observar este sistema es homogéneo. Para saber que condiciones hace falta cumplir para que este sistema sea un sistema compatible y determinado, se calcula la determinante de la matriz.

$$\Delta = \begin{vmatrix} 1 - \eta \cos(\omega N) & -\eta \sin(\omega N) \\ -\eta \sin(\omega N) & 1 + \eta \cos(\omega N) \end{vmatrix} \quad (31)$$

$$\Delta = (1 - \eta^2 \cos^2(\omega N)) - \eta^2 \sin^2(\omega N) \quad (32)$$

$$\Delta = 1 - \eta^2 \quad (33)$$

Por tanto si η toma un valor real diferente de ± 1 , el sistema homogéneo resulta ser un sistema compatible y determinado. En consecuencia $\operatorname{Re}(\omega)$ y $\operatorname{Im}(\omega)$ tomarán valor nulo. En conclusión cualquier polinomio $Q(z)$ con un η diferente de ± 1 tiene las mismas raíces de módulo 1 que $P(z)$.

En el caso de que η tomase los valores ± 1 , el sistema anterior sería compatible e indeterminado. Eso quiere decir que el polinomio $Q(z)$ además de tener entre sus raíces las raíces de módulo unitario de $P(z)$, tendrá otras raíces de módulo unitario. Para calcular esto vamos a estudiar las raíces de módulo unitario de $Q(z)$ para los valores ± 1 .

Estudiamos si existe algún valor w que cumpla el sistema de ecuaciones anterior cuando η toma el valor 1. Debido a que este sistema de ecuaciones es indeterminado basta estudiar la primera ecuación del sistema.

$$\begin{aligned} 0 &= (1 - \eta \cos(\omega N)) \operatorname{Re}(\omega) \\ -\eta \sin(\omega N) \operatorname{Im}(\omega) \end{aligned} \quad (34)$$

$$\begin{aligned} (1 - \eta \cos(\omega N)) \operatorname{Re}(\omega) &= \\ \eta \sin(\omega N) \operatorname{Im}(\omega) \end{aligned} \quad (35)$$

$$\frac{\operatorname{Im}(\omega)}{\operatorname{Re}(\omega)} = \frac{1 - \eta \cos(\omega N)}{\eta \sin(\omega N)} \quad (36)$$

$$\frac{\operatorname{Im}(\omega)}{\operatorname{Re}(\omega)} = \frac{1 - \cos(\omega N)}{\sin(\omega N)} \quad (37)$$

Si se estudia la función $\frac{1 - \cos(\omega N)}{\sin(\omega N)}$ en el

intervalo abierto de valores para w desde 0 hasta $2\pi/N$, se puede advertir que esta función es continua, y que toma todos los valores reales. Eso implica que esta función se corta con la función $\frac{\operatorname{Im}(\omega)}{\operatorname{Re}(\omega)}$ en uno o varios valores de w en ese intervalo. Por tanto $Q(z)$ tiene además de las raíces unitarias de $P(z)$ como raíces,

tendrá una ó más raíces unitarias cuando $\frac{\operatorname{Im}(\omega)}{\operatorname{Re}(\omega)}$

vale 1.

En el caso de que η valga -1 se hace el mismo razonamiento solamente que ahora tenemos la siguiente igualdad. Al igual que en el caso anterior esta otra función también toma todos los valores reales y además es continua en el intervalo abierto desde 0 hasta $2\pi/N$.

$$\frac{\text{Im}(\omega)}{\text{Re}(\omega)} = -\frac{1 + \cos(\omega N)}{\sin(\omega N)} \quad (38)$$

En este caso también se puede afirmar que el polinomio $Q(z)$ presenta más raíces unitarias que las raíces unitarias de $P(z)$. De esta forma quedaría demostrado completamente el teorema.

2.1.2 Teorema 2

Las raíces de $Q(z)$ varían de forma continua para todo valor de η salvo para $\frac{a_0}{a_N}$. En las cercanías de este valor al menos una de las raíces tiende hacia infinito.

Demostración:

El polinomio $Q(z)$ se puede escribir de la siguiente forma.

$$\begin{aligned} Q(z) = & (a_0 - \eta a_N) z^N + \\ & (a_1 - \eta a_{N-1}) z^{N-1} + \dots \\ & \dots + (a_{N-1} - \eta a_1) z + (a_N - \eta a_0) \end{aligned} \quad (39)$$

Igualándolo a cero y haciendo uno el coeficiente de más potencia, se obtiene la ecuación siguiente.

$$z^N + \dots + \left(\frac{a_{N-1} - \eta a_1}{(a_0 - \eta a_N)} \right) z + \left(\frac{a_N - \eta a_0}{(a_0 - \eta a_N)} \right) = 0 \quad (40)$$

Por la formulas de Cardano-Vieta sabemos que el termino independiente es el producto de todas las raíces. Por tanto si variamos el parámetro η y alguna o mas raíces sufren una discontinuidad el término independiente también sufrirá esa discontinuidad. Como el termino independiente solamente sufre una

discontinuidad para el valor $\frac{a_0}{a_N}$, se puede decir que

para ese valor al menos una de las raíces ha evolucionado hacia infinito. Con lo cual demostramos el teorema 2.

3 DEMOSTRACIÓN DE LA PRUEBA O TEST DE JURY

Cuando η vale 0 las raíces de $Q(z)$ tiene las mismas raíces que $P(z)$. Si η vale $\frac{a_N}{a_0}$ el polinomio $Q(z)$ tiene un termino independiente nulo, eso implica que

este polinomio $Q(z)$ tiene una raíz en $z=0$. Por tanto cuando el parámetro η varía desde 0 hasta $\frac{a_N}{a_0}$ al

menos una de la raíces de $P(z)$ ha tenido que evolucionar hasta $z=0$. Sabemos también que una raíz

de $Q(z)$ tiende a infinito si η tiende a $\frac{a_0}{a_N}$ (ver

teorema 2). Finalmente si hacemos variar el parámetro η en el intervalo abierto $(-1,1)$ las raíces del polinomio $Q(z)$ no pueden cruzar la circunferencia de radio unidad, puesto que las raíces de radio unitario de $P(z)$ los son de $Q(z)$, y estas no varían, y $Q(z)$ no logra más raíces unitarias hasta que η toma los valores 1 o -1 . Por tanto ninguna raíz de $Q(z)$ podrá cruzar el círculo unidad dentro del intervalo abierto $(-1,1)$.

1.-Supongamos que $\frac{a_N}{a_0}$ es positivo: Haremos que

η varíe desde 0 hasta 1. Si $\frac{a_N}{a_0}$ es menor que 1, η

tomará ese valor en el intervalo $(0,1)$. Por tanto sabemos que al menos una de las raíces esta dentro del círculo unidad, y por tanto al menos una de la raíces de $P(z)$ será estable. Hay que recordar que las raíces de $Q(z)$ y de $P(z)$ coinciden cuando η tome el valor nulo. En este caso nos quedaremos con el

polinomio $Q(z)$ con η igual a $\frac{a_N}{a_0}$. Debido a que si

bien las raíces de $Q(z)$ no cambian de zona entre el exterior y el interior del círculo unidad, y que además se ha demostrado que una de las raíces es estable, se repetirá la operación de llevar otra raíz a $z=0$. Para hacer esta operación haremos uso del polinomio

$Q(z)/z$ con η igual a $\frac{a_N}{a_0}$. Con cada operación el

polinomio reduce su grado en uno. Si se demuestra que todas las raíces son estables, indicaremos que es sistema es estable.

$$\frac{a_N}{a_0} > 0 \Rightarrow \frac{a_N}{a_0} < 1 \quad (41)$$

Si por el contrario $\frac{a_N}{a_0}$ es mayor que uno, la raíz no

podrá cruzar el círculo unidad. Pero sabremos que una de las raíces está fuera del círculo unidad por que

si $\frac{a_N}{a_0}$ es mayor que uno, $\frac{a_0}{a_N}$ será menor que uno,

y entonces η podrá tomar el valor $\frac{a_0}{a_N}$ ya que está

dentro del intervalo $[0,1)$. Debido a esto una de las raíces irá al infinito. Esto nos lleva a la conclusión de que el sistema es inestable.

2.-Supongamos que $\frac{a_N}{a_0}$ es negativo: Haremos que

η varíe desde 0 hasta -1. Si $\frac{a_N}{a_0}$ es mayor que -1,

η tomará ese valor en el intervalo $(-1,0)$. Por tanto sabemos que al menos una de las raíces está dentro del círculo unidad, y por tanto al menos una de las raíces de $P(z)$ será estable. Hay que recordar que las raíces de $Q(z)$ y de $P(z)$ coinciden cuando η tome el valor nulo. En este caso nos quedaremos con el

polinomio $Q(z)$ con η igual a $\frac{a_N}{a_0}$. Debido a que si

bien las raíces de $Q(z)$ no cambian de zona entre el exterior y el interior del círculo unidad, y que además se ha demostrado que una de las raíces es estable, se repetirá la operación de llevar otra raíz a $z=0$. Para hacer esta operación haremos uso del polinomio

$Q(z)/z$ con η igual a $\frac{a_N}{a_0}$. Con cada operación el

polinomio reduce su grado en uno. Si se demuestra que todas las raíces son estables, indicaremos que el sistema es estable.

$$\frac{a_N}{a_0} < 0 : \frac{a_N}{a_0} > -1 \Rightarrow \text{Raíz estable.} \quad (42)$$

Si por el contrario $\frac{a_N}{a_0}$ es menor que -1, la raíz no

podrá cruzar el círculo unidad. Pero sabremos que una de las raíces está fuera del círculo unidad por que

si $\frac{a_N}{a_0}$ es menor que -1, $\frac{a_0}{a_N}$ será mayor que -1, y

entonces η podrá tomar el valor $\frac{a_0}{a_N}$ ya que está

dentro del intervalo $(-1,0]$. Debido a esto una de las raíces irá al infinito. Esto nos lleva a la conclusión de que el sistema es inestable.

4 CONCLUSIÓN

La condición para que la raíz sea estable será:

$$c = \left| \frac{a_N}{a_0} \right| < 1 \quad (43)$$

Si se calcula de forma recursiva como se indica en la demostración se podrán determinar todos los coeficientes c . Si todos estos coeficientes son menor que 1 el sistema será estable. Si alguno de ellos no cumple la anterior inecuación el sistema se considerará directamente inestable.

Como se puede apreciar este trabajo es una demostración sencilla que se puede presentar a los alumnos de ingeniería. De esta forma los alumnos podrán interpretar los casos extremos que pueden aparecer en dicha prueba.

Referencias

- [1] Y. Bistritz, (1984) Zero location with respect to the unit circle of discrete-time linear system polynomials, Proc. of the IEEE 72 1131-1142.
- [2] M. Benidir and B. Picinbono, (1990) Extended table for eliminating the singularities in Routh's array, IEEE Trans. on Aut. Control 35 218-222.
- [3] F. R. Gantmacher, (1960) The Theory of Matrices, Volume two, Chelsea Publishing Company, New York.
- [4] J. S. H. Tsai and S. S. Chen, (1993) Root distribution of a polynomial in subregions of the complex plane, IEEE Trans. on Aut. Control 38 173-178.
- [5] H. Chapellat, M. Mansour and S. P. Bhattacharyya, (1990) Elementary proofs of some classical stability criteria, IEEE Trans. on Aut. Control 33 232-239.
- [6] A. Lepschy, G. A. Mian and U. Viaro, (1988) A geometrical interpretation of the Routh test, Journal of the Franklin Institute 325,6 695-703.
- [7] G. Meinsma (1995), Elementary proof of the Routh-Hurwitz test, Systems & Control Letters 25 237-242

SISTEMA DE DETECCIÓN AUTOMATIZADO DE GRIETAS EN PIEZAS DE FUNDICIÓN MEDIANTE ULTRASONIDOS Y TÉCNICAS CLÁSICAS DE RECONOCIMIENTO

Cristina Rodríguez González

Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática
E. T. S. de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria
Avda. Los Castros s/n, 39005, Santander, Cantabria, España
E-mail: cristina@teisa.unican.es

M. Fernandez Moreno, L. Alonso Rentería, J. Perez Oria

Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática
E. T. S. de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria
Avda. Los Castros s/n, 39005, Santander, Cantabria, España
E-mail: [monica, alonso, oria]@teisa.unican.es

Resumen

En el presente trabajo se presenta una fase preliminar para obtener un sistema automatizado capaz de distinguir los distintos tipos de fisuras que se pueden encontrar en piezas de fundición a partir de la propagación de ultrasonidos en sólidos (ensayos no destructivos).

Aunque existen otras técnicas para la detección de fisuras los ultrasonidos, debido a su versatilidad, son ampliamente utilizados en aplicaciones con piezas de fundición.

Palabras Clave: ultrasonidos, detección de fisuras, vector de características, clasificador, funciones discriminantes.

1 INTRODUCCION

El origen del uso de los métodos de ensayos no destructivos de materiales surgió de la necesidad de encontrar una técnica que permitiese la exploración interna de un objeto sin que ello conllevara su deterioro [5] [6].

Para la detección de fisuras dentro de los ensayos no destructivos se pueden utilizar diversas técnicas como son el ensayo de líquidos penetrantes, de partículas filtradas, electrizadas o magnéticas, corrientes inducidas o técnicas más convencionales como son los ultrasonidos.

La técnica ultrasónica es la más común debido a que se puede emplear en una gran gama de aplicaciones

y su uso es más sencillo en cuanto a que no exige grandes requerimientos de hardware ni de software.

El objetivo de este trabajo es el de obtener un sistema automatizado de inspección de materiales usando técnicas END que permita la detección de heterogeneidades o fisuras internas en piezas de fundición así como obtener su clasificación según su naturaleza, morfología, orientación, posición o tamaño.

Esta técnica junto con las técnicas clásicas de reconocimiento de objetos permite el diseño de dicho sistema [1] [4].

Según el método de reconocimiento de objetos se puede trabajar con varios enfoques:

- *Aproximación estadística:* Se aplica cuando las formas con las que se trabaja son simples y se pueden caracterizar con un sencillo vector de características. La clasificación se realiza con la ayuda de funciones discriminantes o de decisión y con estudios estadístico-probabilísticos.

- *Aproximación estructural:* Se usa cuando las formas son más complejas y requieren de un vector de características de grandes dimensiones para su representación. En este caso cada forma se considera constituida por varias formas simples entre las cuales existirán relaciones estructurales. La clasificación se realiza con un análisis sintáctico de dichas formas.

En este trabajo se considera el uso de funciones discriminantes para la construcción de clasificadores que permitan catalogar las fisuras. Un clasificador es un sistema de reconocimiento en el cual todas las formas se caracterizan con un vector de características y cuyo funcionamiento se muestra en la Figura 1:

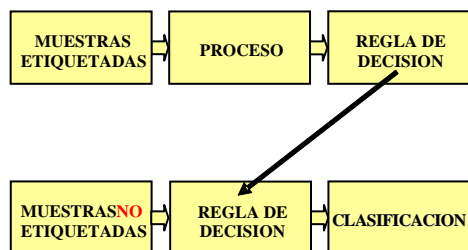


Figura 1: Esquema de un clasificador

En este caso se utiliza el aprendizaje supervisado, inicialmente de cada clase se dispone de una serie de muestras (muestras etiquetadas). Después de un proceso concreto en el cual se hace un estudio sobre las características más adecuadas para la clasificación, se obtiene la regla de decisión. Una vez construida ésta, se hace pasar por ella a muestras de las cuales se desconoce la clase de pertenencia (muestras no etiquetadas) y se obtiene a partir de ella la clasificación.

La selección de características es muy importante porque éstas no se deben elegir al azar sino que deben cumplir unos requisitos concretos [2][3]. Deben tener poder de discriminación, es decir, los valores de las características para objetos que pertenezcan a diferentes clases deben ser distintos; deben ser fiables lo que significa que las características de los objetos de la misma clase deben tomar valores similares. Por último, no debe existir correlación entre las características ya que, en el caso de existir, describirán la misma propiedad pero de forma diferente.

Además, el número de características no debe ser muy grande porque la complejidad de un sistema de reconocimiento aumenta de forma muy significativa según se aumenta el conjunto de características seleccionadas; por tanto, se debe seleccionar un número mínimo de ellas de forma que definan los objetos adecuadamente y que se tenga la menor carga computacional posible para reducir el tiempo de procesado.

2 DESCRIPCIÓN DEL SISTEMA

Para el presente trabajo se ha utilizado el Krautkramer USLT2000, un equipo de exploración de ensayos no destructivos con el que se han capturado los ecos de la pieza objeto de estudio (Figura 2). Con dicho equipo se obtiene la envolvente de dichos ecos.

El sensor que se ha utilizado para las medidas es el MSEB 4-E de la casa Krautkramer que es un palpador de contacto de incidencia normal, de doble cristal (uno emisor y otro receptor) y con una

frecuencia de trabajo de 4 MHz. A la hora de trabajar con él se ha elegido el modo de operación de emisor-receptor.



Figura 2: Equipo de medida.

La pieza de estudio se muestra en la Figura 3, es una probeta de acero cuyas dimensiones son 266 mm de largo, 30 mm de ancho y 38 mm de altura y dispone de una serie de fisuras a lo largo de ella que abarcan el largo de la pieza. Dichas fisuras son de 1 mm de espesor y de diferentes alturas (desde 2 mm a 12 mm en paso de 2 mm).



Figura 3: Sensor en contacto con la pieza.

3 CAPTURA DE ECOS Y SELECCIÓN DE CARACTERÍSTICAS

La forma de proceder para el objetivo de este trabajo es la siguiente:

Se han capturado ecos de las zonas entre fisuras que son los que se van a utilizar como ecos patrón de la clase 1 (sin fisuras), también se han capturado ecos de la zona de fisuras que son los que se van a utilizar para el eco patrón de la clase 2 (con defectos).

La representación gráfica que se obtiene para ambos tipos de ecos, centrándose en la zona que va desde el impulso de emisión hasta el eco de fondo correspondiente al primer rebote se presenta en la Figura 4.

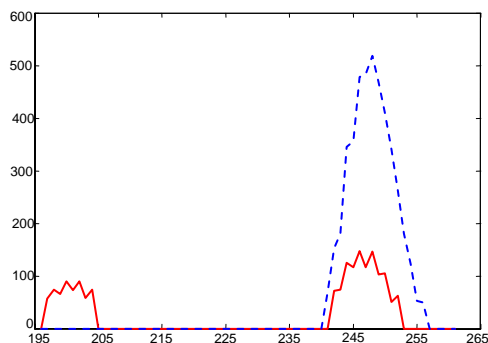


Figura 4: Representación de los dos tipos de ecos

En azul y línea discontinua se muestra un eco de la clase 1 mientras que, en rojo y trazo continuo, se muestra un eco de la clase 2. En este último eco se observa la aparición de un eco debido a un rebote en la fisura y una disminución de la amplitud en el eco de fondo debido a la aparición de dicho defecto.

A la vista de dichos ecos las características que se han calculado son: posición del primer eco (*fondo*), amplitud máxima (*amax*), posición de la muestra máxima (*nmax*), pendiente de subida (*pte*) y área bajo la envolvente del eco (*area*). Una representación gráfica de cómo se obtienen dichas características se muestra en la Figura 5.

Para el valor de la pendiente (1) se ha utilizado la siguiente aproximación:

$$pte = \frac{amax - amp(fondo)}{nmax - fondo} \quad (1)$$

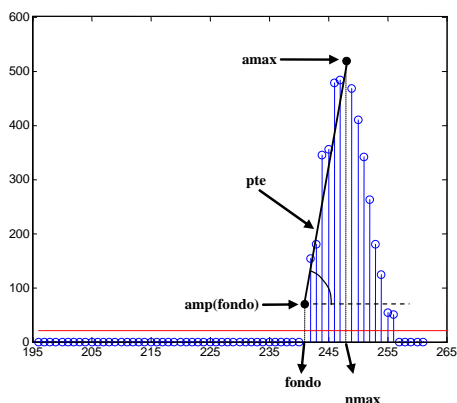


Figura 5: Representación gráfica de las características

De cada clase se han tomado 6 ecos y, los valores de las características para cada uno de ellos, junto con los valores medios se muestran en la Tabla 1 y en la Tabla 2:

Tabla 1: Valores para los ecos de la clase 1.

	fondo	nmax	amax	pte	area
Eco11	242	250	459	48.375	3706
Eco12	242	250	492	52.75	3908
Eco13	242	250	467	49.125	4284
Eco14	243	250	519	64.2857	4471
Eco15	244	250	427	53.3333	2999
Eco16	244	250	446	56.5	3163
Valores medios	243	250	468	54.0615	3755

Tabla 2: Valores para los ecos de la clase 2.

	fondo	nmax	amax	pte	area
Eco21	200	205	99	9.8	1914
Eco22	199	202	89	12.6667	2094
Eco23	200	204	83	3.5	1907
Eco24	199	202	90	12.3334	1655
Eco25	199	204	92	7.2	1770
Eco26	199	202	90	11	1699
Valores medios	199	203	91	9.4167	1840

Estos datos son los que se han utilizado para el diseño de los clasificadores.

4 DISEÑO DEL CLASIFICADOR

En este trabajo se han utilizado dos tipos de clasificadores, cuyo desarrollo teórico se muestra a continuación [1] [4].

4.1 CLASIFICADOR EUCLIDEO

Los problemas de reconocimiento que se resuelven basándose en la distancia Euclídea parten de dos hipótesis; la primera de ellas consiste en que todos los objetos deben estar caracterizados por un único vector de características y, la segunda, indica que se debe disponer de antemano de toda la información necesaria para el diseño del clasificador (conocimiento a priori).

Lo que se ha de tener en cuenta para la deducción del clasificador es que se ha considerado el discriminante lineal más sencillo (caso bidimensional=dos características).

En el problema a resolver se dispone de un espacio de dos dimensiones en el cual se definen dos clases C_1 y C_2 . Cada una de ellas lleva asociada una función discriminante d_i cuya expresión se calcula a partir del vector de características.

Como función discriminante se va a utilizar la distancia Euclídea (2):

$$d_E(\vec{x}, \vec{p}_i) = \|\vec{x} - \vec{p}_i\| = \sqrt{\sum_{j=1}^n (x_j - p_{ij})^2} \quad (2)$$

donde \vec{x} es el vector de características y \vec{p}_i es el centroide o vector prototipo de cada clase (vector media). Uno de los problemas que tiene esta distancia es que no es función lineal pero este hecho se resuelve trabajando con el cuadrado de la misma, la función de decisión (3) que se obtiene es:

$$d_i(\vec{x}) = \vec{x}^T \vec{p}_i - \frac{1}{2} \vec{p}_i^T \vec{p}_i \quad (3)$$

La regla de clasificación para un objeto catalogado por un vector de características \vec{x} vendrá dada por:

$$\text{Si } d_1(\vec{x}) > d_2(\vec{x}) \Rightarrow \vec{x} \in C_1$$

$$\text{Si } d_1(\vec{x}) < d_2(\vec{x}) \Rightarrow \vec{x} \in C_2$$

La zona de incertidumbre (5) se presenta cuando los valores de las dos funciones es el mismo.

$$d_1(\vec{x}) = d_2(\vec{x}) \Rightarrow w_1 x_1 + w_2 x_2 + w_3 = 0 \quad (5)$$

Dicha ecuación corresponde a la ecuación de una recta (6) en el espacio de características que separa dicho espacio en dos zonas cada una de las cuales caracteriza a una de las dos clases.

$$d(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 \quad (6)$$

Teniendo esto en cuenta se obtiene que la regla de clasificación viene dada por:

$$\vec{x} \in C_1 \text{ si } d(\vec{x}) > 0$$

$$\vec{x} \in C_2 \text{ si } d(\vec{x}) < 0$$

4.2 CLASIFICADOR VECINO MÁS PRÓXIMO

En este caso se utiliza el criterio de distancia mínima. Cada una de las clases está representada por un vector prototipo. La regla de decisión utilizada es: un objeto de vector \vec{x} a catalogar se asociará a la clase C_i si la distancia de \vec{x} a \vec{p}_i es la mínima de todas las posibles.

Para el caso de dos clases se tiene el siguiente clasificador:

$$\vec{x} \in C_1 \text{ si } D(\vec{x}, \vec{p}_1) < D(\vec{x}, \vec{p}_2)$$

$$\vec{x} \in C_2 \text{ si } D(\vec{x}, \vec{p}_1) > D(\vec{x}, \vec{p}_2)$$

donde D representa la función distancia utilizada y debe ser una distancia métrica.

Para este trabajo se ha utilizado la distancia Euclídea indicada anteriormente (2).

4.3 RESULTADOS

Para la obtención de los distintos clasificadores se ha utilizado siempre como primera característica discriminante el *fondo* (x_1) y se da la opción de elegir una de las restantes (*nmax*, *amax*, *pte*, *area*) como segunda característica discriminante (x_2).

4.3.1 Clasificador Euclídeo

A continuación se muestran en la Tabla 3 los resultados obtenidos para las fronteras de decisión de cada caso:

Tabla 3: Fronteras de decisión para el clasificador Euclideo

<i>fondo y amax</i>	$44x_1 + 377x_2 - 115095.5 = 0$
<i>fondo y nmax</i>	$44x_1 + 47x_2 - 20369.5 = 0$
<i>fondo y pte</i>	$44x_1 + 44.645x_2 - 11140.987 = 0$
<i>fondo y area</i>	$44x_1 + 1915.333x_2 - 5367869 = 0$

La representación gráfica de dichas fronteras en cada espacio de características se muestra en la Figura 6.

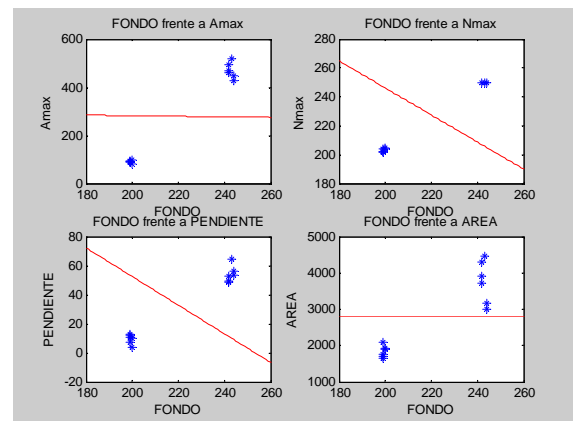


Figura 6: Fronteras de decisión para cada espacio de características

4.3.2 Clasificador vecino más próximo

Para este clasificador lo que interesa es conocer para cada caso los valores de los centroides correspondientes a cada clase que se muestran a continuación en la Tabla 4.

Tabla 4: Centroides de cada clase para el clasificador vecino más próximo

	Clase 1	Clase 2
fondo y amax	(243 468)	(199 91)
fondo y nmax	(243 250)	(199 203)
fondo y pte	(243 54.0615)	(199 9.4167)
fondo y area	(243 3755)	(199 1840)

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 7.

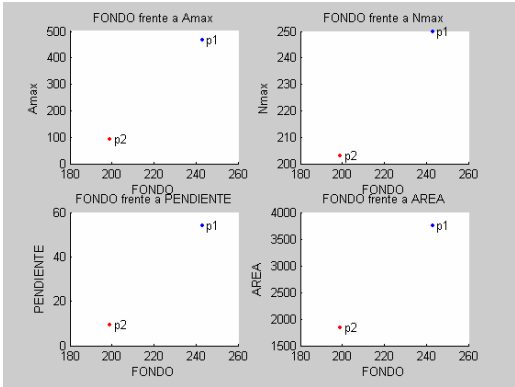


Figura 7: Centroides de cada clase en cada espacio de características

5 PRUEBA DEL CLASIFICADOR

Para comprobar la eficacia de cada clasificador se han capturado en la pieza una serie de ecos al azar. De cada uno de ellos se han obtenido los correspondientes vectores de características cuyos valores se muestran en la Tabla 5.

Tabla5: Vectores de características para los ecos de prueba

	fondo	nmax	amax	pte	area
prueba1	242	250	453	49.125	3787
prueba2	200	205	84	3.8	1841
prueba3	243	251	499	54	4391
prueba4	242	250	444	47.875	3660
prueba5	199	204	89	7.6	2881
prueba6	201	205	90	3.75	2086

La pertenencia a cada una de las clases se conoce previamente y es la siguiente: los ecos 1,3,4 pertenecen a la clase 1 y los ecos 2,5,6 pertenecen a la clase 2.

A este conjunto de ecos se les han aplicado los dos clasificadores para comprobar si verdaderamente funcionan. Los resultados que se han obtenido se muestran a continuación.

5.1 CLASIFICADOR EUCLIDEO

5.1.1 Caso fondo y nmax

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 8.

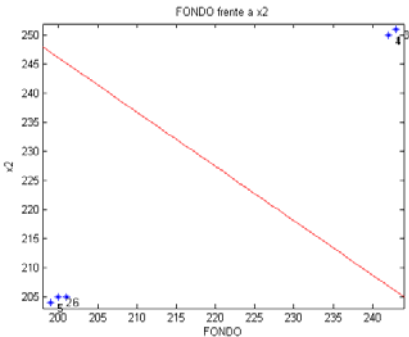


Figura 8: Clasificador Euclídeo (fondo-nmax)

5.1.2 Caso fondo y amax

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 9.

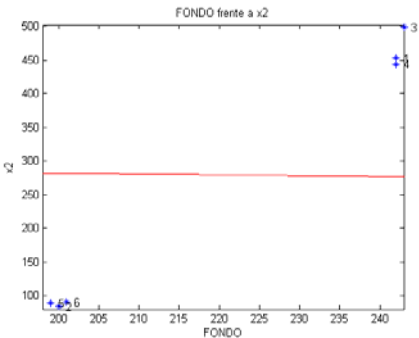


Figura 9: Clasificador Euclideo (fondo-amax)

5.1.3 Caso fondo y pte

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 10.

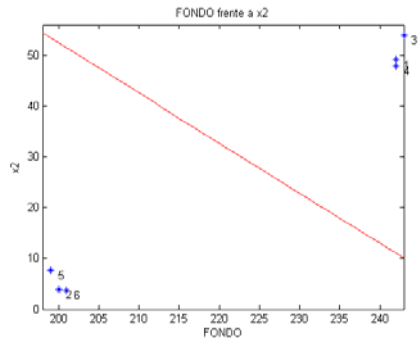


Figura 10: Clasificador Euclideo (fondo-pt)

5.1.4 Caso fondo y area

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 11.

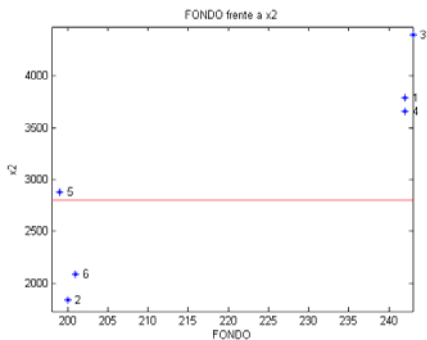


Figura 11: Clasificador Euclideo (*fondo-area*)

5.2 CLASIFICADOR VECINO MAS PROXIMO

5.2.1 Caso fondo y nmax

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 12.

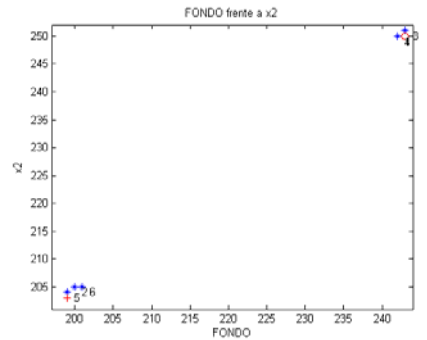


Figura 12: Vecino más Próximo (*fondo-nmax*)

5.2.2 Caso fondo y amax

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 13.

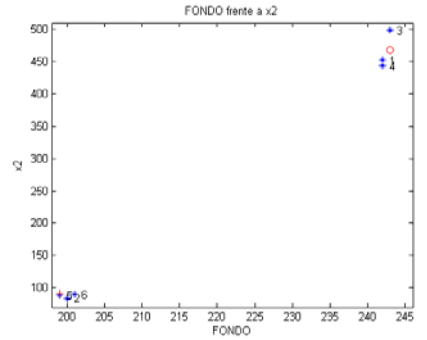


Figura 13: Vecino más Próximo (*fondo-amax*)

5.2.3 Caso fondo y pte

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 14.

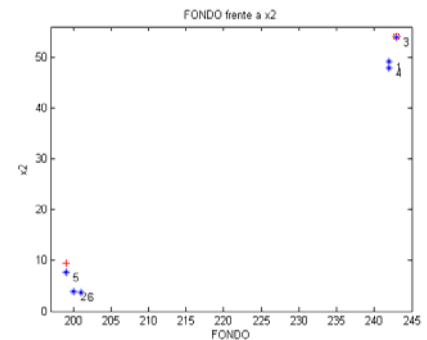


Figura 14: Vecino más Próximo (*fondo-pte*)

5.2.4 Caso fondo y area

La representación gráfica de los objetos en el espacio de características se muestra en la Figura 15.

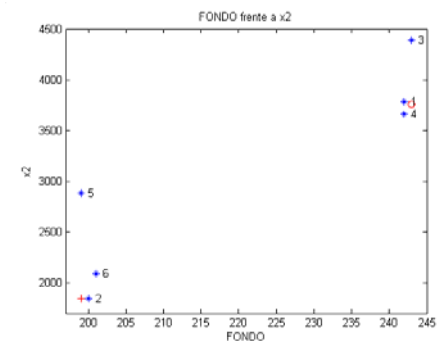


Figura 15: Vecino más Próximo (*fondo-area*)

5.3 RESULTADOS

Para el clasificador Euclideo en los tres primeros casos se obtiene la siguiente clasificación:

```
----- DISCRIMINADOR EUCLIDEO MODIFICADO -----
x1=FONDO      x2
-----
clase1 = 1      3      4
clase2 = 2      5      6
```

y, para el último caso:

```
----- DISCRIMINADOR EUCLIDEO MODIFICADO -----
x1=FONDO      x2
-----
clase1 = 1      3      4      5
clase2 = 2      6
```


Para el clasificador Vecino Mas Próximo en los tres primeros casos se obtiene la siguiente clasificación:

DISTANCIA EUCLIDEA	
x1=FONDO	x2
Objeto	Clase
1	1
2	2
3	1
4	1
5	2
6	2

y, para el tercer caso:

DISTANCIA EUCLIDEA	
x1=FONDO	x2
Objeto	Clase
1	1
2	2
3	1
4	1
5	1
6	2

A la vista de las clasificaciones obtenidas para cada uno de los dos clasificadores, puede deducirse que cualquiera de los 3 primeros casos (*fondo-nmax*, *fondo-amax*, *fondo-pte*) se puede utilizar como sistema de detección de fisuras o grietas en las piezas de fundición porque se tiene un espacio de características que cumplen las propiedades adecuadas de discriminación, fiabilidad e independencia que se indicaron en la introducción del trabajo y son capaces de distinguir entre clases.

Sin embargo, el último caso (*fondo-area*) aunque aparentemente se tenía un espacio de características que cumplía dichas propiedades al probar con los ecos de prueba no es capaz de discriminar entre ellos.

6 CONCLUSIONES

Dado el número reducido de clases que se han utilizado los dos clasificadores seleccionados muestran un comportamiento eficiente a la hora de detectar las fisuras en las piezas de fundición.

Si el problema se traslada a un caso más real, en el cual haya que distinguir entre varios tipos de fisuras, aumenta el número de clases por lo que aumenta su dificultad y, por tanto, la complejidad del software de detección.

En este caso, entre uno u otro clasificador, el que obtendrá mejores resultados de una forma menos complicada será el del vecino más próximo porque la regla de decisión para el clasificador Euclideo es

mucho más difícil de deducir ya que aumenta el número de fronteras de decisión y, por tanto, a la hora de obtener el sistema automatizado éste es más complicado de programar.

Referencias

- [1] Escalera Hueso, Arturo de la, (2001) Visión por computador. Fundamentos y métodos. Ed. Prentice-Hall, Madrid
- [2] Dubes R., Jain A.K., 1976. Clustering Techniques: The User's Dilemma. Pattern Recognition, vol.8, pp. 247-260.
- [3] Llata J.R., Sarabia E.G., Oria J.P., Pattern Recognition with Ultrasonic Sensors: a Neural Network Evaluation. Sensor Review, Vol 21 N°1, pp 45-57. 2001.
- [4] Maravall Gómez-Allende, Darío, (1993) Reconocimiento de formas y visión artificial. Ed. Ra-Ma, Madrid
- [5] Ramírez Gómez, Francisco; Introducción a los métodos de ensayos no destructivos de control de la calidad de los materiales. (3ª Edición) Instituto Nacional de Técnica Aeroespacial, Madrid.
- [6] Krautkrämer J. y Krautkrämer H. Ultrasonic Testing of Materials. Ed. Springer-Verlag, Berlin, 4ª edición, 1990

COLIBRI: Vehículo Aéreo Autónomo Guiado por Visión para Inspección y Vigilancia

Luis Mejías y Pascual Campoy
Grupo de Visión por Computador
ETS Ingenieros Industriales
Universidad Politécnica de Madrid
Madrid, España
{lmejias,campoy}@etsii.upm.es

Resumen

El uso de sensores visuales en robots aéreos para realizar tareas como evasión de obstáculos, seguimiento de objetivos o detección de características salientes, permite a tales robots servir de visores inteligentes para aplicaciones de inspección, patrullaje y monitoreo, reconstrucción digital aérea, etc. Más aún, el uso de la visión por computador puede reducir la incertidumbre e incrementar la versatilidad y desempeño general cuando se realizan tareas robotizadas. Este trabajo aborda la problemática del control servo visual de un vehículo aéreo autónomo en espacios exteriores, en especial de un helicóptero autónomo. El siguiente trabajo propone técnicas de visión por computador que permiten a un vehículo aéreo realizar maniobras mientras se hace seguimiento de características en entornos donde el GPS tiene fallos de recepción (usualmente en zonas urbanas) o simplemente permiten alinear el vehículo con un objetivo dado. Se investigan técnicas de control servo visual que usan directamente la posición de las características en la imagen para generar referencias de velocidad para control de vuelo, es decir, la tarea se especifica en el espacio de trabajo del sensor. Esta estrategia no requiere procedimientos calibración de la cámara o reconstrucción 3D de la escena, los cuales están sujetos a errores y demandan una alta capacidad de computacional.

Palabras clave: control visual, helicóptero autónomo, navegación basada en visión, detección y seguimiento visual.

1. Introducción

Los vehículos aéreos autónomos (UAV, de sus siglas inglesas Unmanned Aerial Vehicles. Nomenclatura internacional adoptada en el presente documento) han sido un área de investigación muy activa durante los últimos años. El uso de UAV en aplicaciones civiles de patrullaje y monitoreo continúa en crecimiento debido en parte por la evolución y reducción en el coste de los sistemas visión. La visión para el control de un UAV implica realizar investigación en diversos campos como la

detección y seguimiento de objetos, estimación de posición, fusión sensorial con GPS y medidas inerciales, y modelado y control de sistemas multivariados no lineales. Un helicóptero autónomo posee propiedades que lo hacen la plataforma idónea para tareas de inspección y monitoreo. Su inherente habilidad de volar a baja velocidad, estacionariamente, lateralmente y realizar maniobras en espacios reducidos lo hacen la plataforma adecuada para tales tareas.



Figura 1: Plataforma aérea COLIBRI durante prueba experimental con líneas eléctricas.

La plataforma aérea presentada (figura 1) combina técnicas de visión por computador con un control de bajo nivel para lograr el control visual de un UAV por medio del seguimiento visual de características en la imagen. El sistema de visión actúa como un controlador de alto nivel enviando consignas de velocidad al control de vuelo, el cual es responsable de un control robusto y estable. El resultado es un algoritmo de seguimiento visual de características que controla los desplazamientos de un helicóptero autónomo en espacios exteriores.

2. Trabajos Relacionados

El estudio de los helicópteros autónomos tiene su origen décadas atrás. El modelado dinámico ha sido ampliamente documentado, en particular en [23], [11] donde se presenta un estudio en detalle sobre el modelo aerodinámico y análisis de estabilidad. Dada la inestabilidad propia de los hel-

icópteros los primeros trabajos de investigación se enfocaron en el diseño de controladores estables. En [29] se documenta el diseño de un control retroalimentado robusto basado en H_∞ , al igual en [27] se presenta un control Fuzzy para un Yamaha R-50.

Entre los primeros sistemas documentados de navegación para helicópteros autónomos se encuentra [6]. Este sistema presentaba la particularidad de solo usar GPS como sensor principal para la navegación sustituyendo a la Unidad de Medida Inercial (IMU), la cual históricamente se había usado como sensor principal. Este sistema estaba dotado de un GPS que consistía de un oscilador principal y 4 receptores (usando desplazamiento en fase) con 4 antenas colocadas en puntos estratégicos sobre el helicóptero con lo que se conseguía su posición, velocidad, actitud e información angular. El helicóptero descrito en [21] posee una arquitectura jerárquica de módulos de control. Módulos de bajo nivel se encargan de tareas reflexivas, de rápida respuesta como control de bajo nivel, mientras los módulos de alto nivel se encargan de tareas de planificación y navegación. El control se ha diseñado con leyes de control lineal usando controladores PID. Entre los sensores que posee están GPS, IMU, Magnetómetro, Sonar, etc.

En [26] se presenta un sistema de control jerárquico usado en el helicóptero: BErkeley AeR-robot project (BEAR) [4]. La identificación del sistema se usa para generar el modelo del helicóptero, y basándose en este modelo se diseñan las leyes de estabilización. El controlador consta de 3 lazos: 1) un lazo interno de control de actitud, 2) un lazo intermedio de control de velocidad, y 3) un lazo externo de control de posición. Las redes neuronales también han servido para diseño e implementación de controladores de vuelo, siendo el Instituto Tecnológico de Georgia [28] el que más aportes ha hecho en esta técnica. Entre los trabajos más importantes de este grupo se encuentran [12], [13], [14] y [22]. Recientes logros en el control y modelado han permitido extender las capacidades de vuelo, llegando a realizar maniobras acrobáticas. Un ejemplo de este sistema es el helicóptero descrito en [7] y [8].

Al hacer referencia al control visual, y en especial al de vehículos autónomos aéreos, se hace referencia al uso de la información visual del procesamiento de imágenes para el control de velocidad, posición absoluta o relativa, o para el control de la orientación de un robot aéreo. Al igual que la literatura sobre control visual contempla el término *camara-en-mano* para ciertas configuraciones en robots articulados, la misma puede usarse para el caso de robots aéreos [15]. Uno de los primeros

helicópteros autónomos guiados por visión se describe en [3]. Este vehículo combina las lecturas GPS con un sistema de visión, con la finalidad aumentar la precisión de la estimación de estado para la navegación. El sistema de visión consiste de un procesador DSP que proporciona medidas de posición, velocidad y actitud a frecuencias del orden de 10ms, que combinado con las lecturas del GPS y IMU aumenta la precisión en la estimación de la actitud y la posición del helicóptero.

El control visual también se ha aplicado a vehículos en miniatura, en el caso de un helicóptero (HMX-4) de cuatro rotores [1], [2] donde la visión se usa para determinar la disposición del helicóptero y para la detección de objetos en tierra. En el caso de aviones en miniatura [5] la detección y localización del horizonte se usa para el control lateral de un mini UAV.

En el área del aterrizaje autónomo basado en visión, recientemente en [20] la detección de un patrón conocido usando visión por computador y la fusión de esta información con las medidas inerciales permite aterrizar este helicóptero en caso de no disponibilidad de GPS. En [10] técnicas de visión por computador son usadas para recobrar y detectar aquellas áreas seguras para aterrizar en terrenos desconocidos. Anteriormente, la visión artificial se usó para aterrizar autónomamente un helicóptero usando dos estrategias diferentes [24] y [25], pero con la particularidad de ser la visión por computador la fuente principal de información.

En el caso de la navegación 3D basada en visión Habrar [9] propone una técnica de evasión de obstáculos basada en visión por computador combinando flujo óptico y visión estereoscópica. Experimentos demuestran las ventajas de combinar ambas técnicas, las cuales a su vez combinadas con un planificador de trayectorias basado en mapas probabilísticos permite la navegación de un helicóptero autónomo en entornos urbanos.

3. Control de vuelo

El control de vuelo está compuesto por varios lazos de control en cascada para controlar la actitud, velocidad y posición. Cada controlador genera las referencias al controlador siguiente, siendo el control de actitud el que genera las consignas finales a los servos del helicóptero.

Los diferentes lazos de control mencionados anteriormente leen el estado de las variables del estimador de estado y estabilizan el helicóptero en un punto de operación basándose en controladores PID desacoplados. La maniobrabilidad del vehículo autónomo considerado, el helicóptero, como se mencionó anteriormente puede ser contro-

lada modificando los ángulos de actitud, es decir, actuando sobre el *roll*, *pitch* y *yaw*. El control de actitud lee los valores de roll, pitch y yaw, y genera las consignas necesarias para estabilizar la actitud del helicóptero. El control del roll y pitch se lleva a cabo mediante dos controladores PD, el control del yaw se hace mediante un controlador PID. Las ecuaciones que rigen el control de actitud se muestran a continuación:

$$\delta_{lat} = K_p(\phi_d - \phi) + K_d \frac{(\phi_d - \phi)}{dt} \quad (1)$$

$$\delta_{lon} = K_p(\theta_d - \theta) + K_d \frac{(\theta_d - \theta)}{dt} \quad (2)$$

$$\delta_t = K_p(\psi_d - \psi) + K_i \int (\psi_d - \psi) dt + K_d \frac{(\psi_d - \psi)}{dt} \quad (3)$$

donde: δ_{lat} , δ_{lon} and δ_t son los comandos de pitch cíclico latitudinal, pitch cíclico longitudinal y colectivo de rotor de cola, respectivamente. Los valores de K_p , K_i and K_d las ganancias proporcional, integral y derivativa asociadas a cada controlador.

El control de velocidad, el cual genera referencias de roll, pitch (al control de actitud) y colectivo de rotor principal, está implementado mediante un control PI. El controlador lee los valores de velocidad del estimador de estado y genera: ϕ_d y θ_d al control de actitud lo que desplaza el vehículo lateral y longitudinalmente a una velocidad dada, respectivamente. La velocidad vertical que se controla actuando sobre el colectivo de rotor principal, se puede modificar por medio del control de velocidad o posición dependiendo del modo de operación. Cuando el vehículo se mueve hacia una coordenada GPS dada, el control de posición genera consignas de colectivo principal basándose en la altura actual y la deseada. Cuando se está en vuelo estacionario o se opera en modo velocidad, el control de velocidad es capaz de recibir referencias externas generando consignas de colectivo principal directamente. Este modo de operación es particularmente útil cuando se realiza control visual.

$$\phi_d = vx_r = K_p(vx_d - vx) + K_i \int (vx_d - vx) dt \quad (4)$$

$$\theta_d = vy_r = K_p(vy_d - vy) + K_i \int (vy_d - vy) dt \quad (5)$$

$$\delta_m = K_p(vz_d - vz) + K_i \int (vz_d - vz) dt \quad (6)$$

donde: vx_r , vy_r and δ_m son la referencias de velocidad longitudinal, velocidad latitudinal y colectivo de rotor principal, respectivamente. Los valores de K_p y K_i son las ganancias proporcional e integral asociadas a cada controlador. Como se mencionó anteriormente en la configuración de control en cascada cada controlador genera los setpoints (referencias) al controlador siguiente, por esta razón las ecuaciones 4 y 5 son los valores deseados de entrada al controlador de actitud, mientras que la ecuación 6 es el valor del comando de colectivo del rotor principal.

El control de posición se implementa mediante un controlador PID. Cuando se vuela en dirección a una coordenada dada o se está en vuelo estacionario, el controlador lee las coordenadas GPS deseadas y la actual, y genera las referencias de velocidad al control de velocidad del vehículo y colectivo de rotor principal.

$$vx_d = x_r = K_p(x_d - x) + K_i \int (x_d - x) dt + K_d \frac{(x_d - x)}{dt} \quad (7)$$

$$vy_d = y_r = K_p(y_d - y) + K_i \int (y_d - y) dt + K_d \frac{(y_d - y)}{dt} \quad (8)$$

$$vz_d = z_d = K_p(z_d - z) + K_i \int (z_d - z) dt + K_d \frac{(z_d - z)}{dt} \quad (9)$$

De manera similar, al control de velocidad el control de posición genera las referencias al controlador siguiente. El controlador enseguida es el de velocidad, por esta razón la ecuación 7 y 9 se pueden escribir como las referencias de velocidad. Sin embargo, dependiendo del modo de operación el control de posición puede generar directamente la consigna de pitch colectivo de rotor principal influyendo sobre la altitud del helicóptero.

4. Integración con referencias visuales

La integración de la visión por computador en el control de vuelo implica otros campos de investigación como la detección y seguimiento de patrones, estimación de posición por flujo óptico, navegación inercial, GPS, modelado no lineal, etc. Durante el procesamiento de imágenes, al realizarse la detección y seguimiento en el plano de la imagen, las referencias más apropiadas que puede generar esta etapa son de velocidad en el plano de la imagen, lo que representa a su vez las

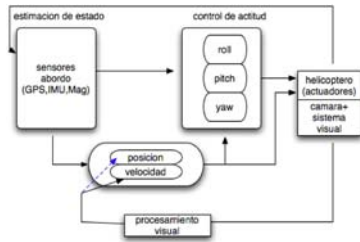


Figura 2: Diagrama general de control. Se muestra la configuración desacoplada del sistema de visión

referencias de velocidad al control de vuelo en el sistema de coordenadas del helicóptero.

La configuración actual propuesta (figura 2) del sistema es desacoplado y basado en imagen, es decir, la señal de error a corregir es la posición de la característica en la imagen. En la mayor para de las aplicaciones derivadas de esta propuesta, la visión se usa para alinear el UAV con el objeto de interés usando referencias de velocidad en el plano de la imagen, por lo que la convergencia de esta señal a cero valida la configuración propuesta.

4.1. Capa de interconexión entre procesos

Esta capa hace uso de una arquitectura conocida como es la cliente-servidor. Esta arquitectura esta basada en mensajes TCP/UDP y esta orientada a que los procesos embebidos se ejecuten a bordo del vehículo. En la sección 5 se describe como son generadas las referencias visuales de velocidad del plano de la imagen. Su integración con el control de vuelo se lleva a cabo mediante mensajes, a través de una capa de software definida por una API de comunicaciones. Esta capa permite la interconexión de diferentes procesos externos simultáneamente.

El esquema propuesto tiene la ventaja de ser enormemente flexible, dado que varios procesos externos pueden inter-actuar entre ellos y con el control de vuelo a través de la capa de conexión, vease figura 3. En el caso del uso de la visión por computador para el control visual, dicha flexibilidad viene del hecho de que varios sub-procesos de visión, y entiendase por sub-procesos como, algoritmos dedicados a tareas y/o aplicaciones específicas, pueden inter-actuar a la vez. En el caso de requerir algoritmos de propósito especial, el esquema sigue siendo válido ya que solo requerirá el cambio interno del proceso de visión.

Las funcionalidades básicas de esta capa se definen como:

- Definir un conjunto de mensajes para identi-

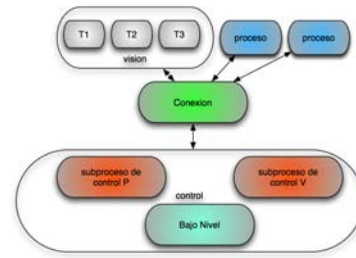


Figura 3: Esquema de procesos. Procesos externos (nivel superior) inter-actúan con el control a través de una capa de conexión

ficar el tipo de información, así como el destino del mensaje. Mensajes y estructuras de datos son definidas, y luego enviados a la capa de interconexión. Diferentes mensajes son creados para el control de vuelo como: control de velocidad, control de posición, orientación, actitud, estado del helicóptero, etc. Para procesos externos como: tipo de información a enviar y recibir, desde y hacia los procesos externos.

- Conmutar y enrutar los mensajes dependiendo del tipo de información contenida. Por ejemplo, la capa puede cambiar entre control de velocidad o posición dependiendo del tipo de mensaje recibido desde un proceso externo.

5. Posicionamiento basado en visión. Definición de la tarea de control visual

La tarea del control visual se encarga de alinear el vehículo con respecto a un objetivo basándose solo en la información visual 2D que extrae de la escena. A continuación se define como se generan las señales de referencias de velocidad que son enviadas al control de vuelo para realizar el control visual. Según el esquema de control mostrado en la figura 2 se definen principalmente tres maniobras que son controladas visualmente, el control de posición lateral, vertical y longitudinal. La figura 4 muestra la disposición de la cámara durante el control visual de la maniobras. Cabe destacar que el control visual lateral del vehículo puede ser llevado a cabo utilizando ambas configuraciones.

5.1. Control visual lateral

Para el control de la velocidad lateral, y por consiguiente, de la posición lateral del helicóptero se puede usar una cámara colocada en la parte frontal del vehículo o en la parte inferior del mismo. En el caso ilustrado en la figura 5(a) la cámara

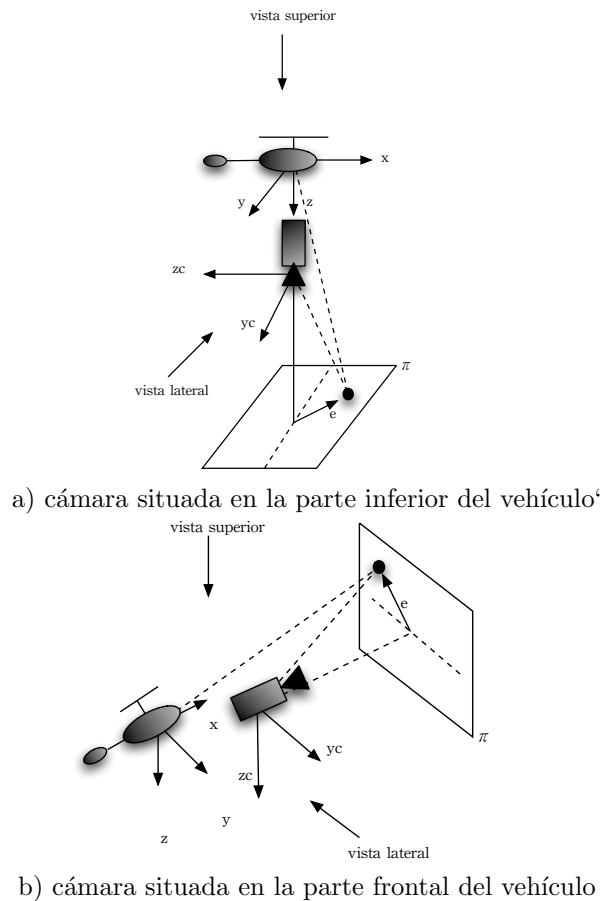


Figura 4: Disposición del helicóptero y la cámara, cámara inferior y cámara frontal. La señal de error se indica en forma vectorial, para expresar que consta de dos componentes de referencia de velocidad en la dirección j e i de la imagen.

esta situada en la parte inferior. La tarea de control visual se define como aquella donde los ángulos α y β se conducen a cero. Pero debido a que en el cálculo de dichos ángulos son introducidos muchos parámetros de incertidumbre, derivados de la calibración y condicionados por la elección de un modelo de cámara más complejo, se escoge como parámetro para la generación de la señal de referencia de la tarea de control visual la coordenada i de la posición de la características en la imagen. Un razonamiento análogo se aplica en el caso en el que la cámara esta situada en la parte frontal del vehículo, figura 5(b).

En ambas configuraciones se usa la coordenada i de la característica en la imagen, siendo las referencias de velocidad lateral válidas para ambas. Estas referencias vienen dadas por la ecuación 10.

$$vy_r = k_l \frac{(i - \frac{w}{2})}{w/2} \quad (10)$$

Donde w es el ancho de la imagen e i la posición

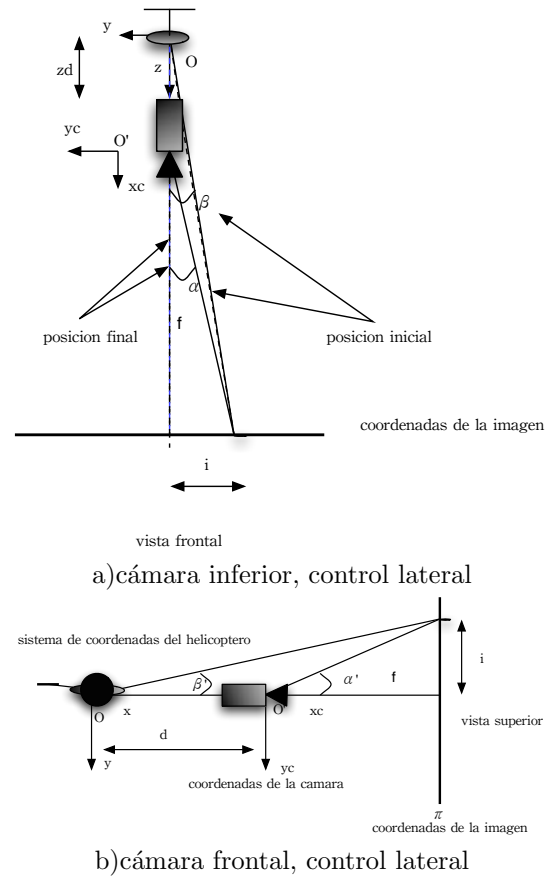


Figura 5: Disposición helicóptero-cámara: control visual de posición lateral

de la característica detectada en la imagen. k_l se usa como ganancia del sistema visual de referencias laterales. Si su valor es igual 1, las referencias varían entre -1 y 1 (m/s). Si es diferente entonces su valor será k_l veces ± 1 (m/s).

Regresando a la figura 5, de ella se puede extraer:

$$\tan \alpha = \frac{i - w/2}{f} \quad (11)$$

$$\tan \beta = \frac{i - w/2}{f + z_d} \quad (12)$$

$$\tan \alpha' = \frac{i - w/2}{f} \quad (13)$$

$$\tan \beta' = \frac{i - w/2}{f + d} \quad (14)$$

donde f es la distancia focal de la cámara, d y x_d la distancia que separa ambos sistemas de coordenadas, respectivamente. En la práctica d y z_d se aproximan a cero colocando la cámara lo más fiel posible al centro de masa del vehículo. Resultando en:

$$\tan \alpha = \tan \beta \quad (15)$$

$$\tan \alpha' = \tan \beta' \quad (16)$$

El control visual tiene la función de hacer que la componente i de la señal de error tienda a $\frac{w}{2}$ (característica centrada en la imagen) con velocidad negativa, esto es, a medida que el vehículo se alinea con el objetivo, $i \rightarrow \frac{w}{2}$, esto genera referencias positivas de velocidad al helicóptero y los ángulos α' y β' tienden a cero, causando a su vez $vy_r \rightarrow 0$. Por lo tanto, al ser los ejes longitudinales de la cámara y el helicóptero solidarios, el helicóptero se mueve lateralmente por medio de las referencias de velocidad dadas por la ecuación 10.

5.2. Control visual vertical

Por medio de un razonamiento análogo al anterior, la velocidad y la posición vertical del helicóptero, son controladas por las referencias de velocidad vertical generadas a partir de la imagen. Estas referencias vienen dadas por la ecuación 17.

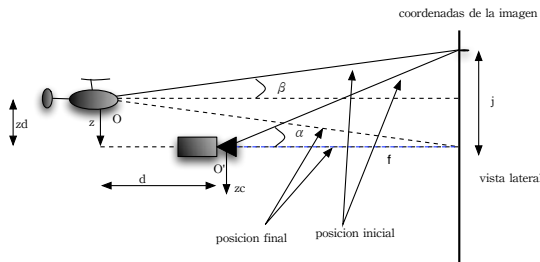


Figura 6: Disposición helicóptero-cámara: control visual de posición vertical (altitud)

$$vz_r = k_v \frac{(j - \frac{h}{2})}{h/2} \quad (17)$$

Donde h es el alto de la imagen y j la posición de la característica detectada en la imagen. Análogamente al caso anterior k_v es la ganancia de sistema visual de referencias verticales.

De figura 6, se puede extraer:

$$\tan \alpha = \frac{j - h/2}{f} \quad (18)$$

$$\tan \beta = \frac{j - z_d - h/2}{f + d} \quad (19)$$

A medida que el control visual corrige la posición del vehículo, la característica tiende a centrarse en la imagen, luego la componente j de la señal de error ($j \rightarrow \frac{h}{2}$), y el ángulo α tiende a cero. No

así, el ángulo β que tiende a $\frac{-z_d}{f+d}$. Esto conlleva a que cuando la característica está centrada en la imagen, verticalmente el vehículo no está alineado con el objetivo, debido al factor z_d . Físicamente se ha encontrado que esta distancia es muy pequeña, por lo que en la práctica no se distingue su efecto y se desprecia.

5.3. Control visual longitudinal

De manera similar a los planteamientos anteriores, la velocidad y la posición longitudinal del helicóptero, serán controladas por las referencias de velocidad generadas a partir de la imagen. Estas referencias vienen dadas por la ecuación 20. Extrayendo de la figura 7, la vista lateral:

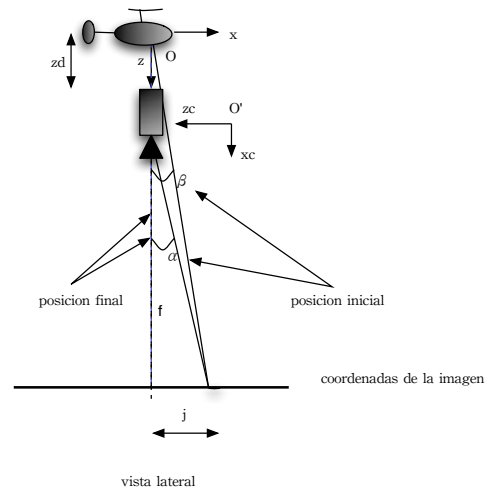


Figura 7: Disposición helicóptero-cámara: control visual de posición longitudinal

$$vx_r = -k_L \frac{(j - \frac{h}{2})}{h/2} \quad (20)$$

donde h es el alto de la imagen y j la posición de característica detectada en la imagen. El signo menos viene del hecho que ambos sistemas de coordenadas son opuestos, es decir, el eje ZC y el eje x tienen sentido contrarios. Análogamente al caso anterior k_L es la ganancia de sistema visual de referencias longitudinales.

De figura 7, se puede extraer:

$$\tan \alpha = \frac{j - h/2}{f} \quad (21)$$

$$\tan \beta = \frac{j - h/2}{f + z_d} \quad (22)$$

A medida que la característica se centra en la imagen la componente j tiende a $\frac{h}{2}$, lo que conlleva a que los ángulos α y β tienden a cero. Con lo que se verifica la convergencia del control longitudinal

5.4. Aplicaciones Experimentales

De entre la varias aplicaciones que se han conseguido con esta plataforma se encuentran

1. Detección y seguimiento de características externas en entornos urbanos. Con el objetivo de la validar la propuesta de control visual anteriormente mencionada, se ha simulado una inspección sobre una fachada de un edificio siguiendo visualmente características elegidas por un usuario en tierra.
2. Visión estereoscópica para la navegación de un vehículo aéreo autónomo. Usando un sistema estereoscópico se consigue calcular la altura y el desplazamiento del sistema detectando y siguiendo puntos salientes en la imagen
3. Planificación de trayectorias usando splines. Dada una serie de coordenadas GPS, el sistema calcula la trayectoria óptima mas adecuada de forma que el UAV visite cada punto siguiendo una ruta continua.
4. Detección y seguimiento de líneas eléctricas de alta tensión. Usando una técnica de detección de segmentos de líneas se encuentran las líneas sobre la imagen, y a continuación el sistema visual busca los objetos de interés o componentes, como aisladores, sobre la línea de manera de inspeccionarlos visualmente.

Algunas de estas aplicaciones y sus resultados mas significativos se describen en [16], [17], [18] y [19].

6. Conclusiones

El tema del control y la navegación visual de vehículos aéreos autónomos ha suscitado un importante interés durante los últimos años y sigue motivando en la actualidad la actividad investigadora de una numerosa comunidad científica. Gran parte de esta actividad se ha centrado en el área de control, se han propuesto y desarrollado técnicas para estabilizar y maniobrar helicópteros autónomamente dada una trayectoria de vuelo. Sin embargo el área de control visual de UAV esta en pleno auge y desde años recientes se viene popularizando el uso de vehículos aéreos no tripulados para inspección, monitoreo y vigilancia visual, lo que a su vez ha llevado a los investigadores del área visión por computador ha desarrollar técnicas que, usando la información que el vehículo adquiere a través de sensores visuales, pueda influir sobre la trayectoria o plan de vuelo acorde con el procesamiento visual de la escena. El presente trabajo

ha mostrado una estrategia de control visual aplicado a un helicóptero autónomo, con aplicaciones potenciales reales en la industria.

Agradecimientos

El trabajo presentado ha sido financiado por el MCYT bajo proyecto DPI2004-06624. Los autores desean agradecer a Jorge León por el apoyo prestado al realizar las pruebas experimentales.

Referencias

- [1] ALTUĞ, E. *Vision Based Control of Unmanned Aerial Vehicles with Applications to an Autonomous Four Rotor Helicopter*. PhD thesis, Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, December 2003.
- [2] ALTUĞ, E., OSTROWSKI, J. P., AND TAYLOR, C. J. Control of a quadrotor helicopter using dual camera visual feedback. *International Journal of Robotics Research* 24, 5 (May 2005), 329–341.
- [3] AMIDI, O. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Robotics Institute, Carnegie Mellon University, 1996.
- [4] BEAR. University of California at Berkeley. BErkeley AeRobot Project. <http://robotics.eecs.berkeley.edu/bear>, 2006.
- [5] CAUSEY, R. S. A lateral vision-based control autopilot for micro air vehicles using a horizon detection approach. Master's thesis, University of Florida. Depart. of Mechanical and Aerospace Engineering, 2003.
- [6] CONWAY, A. R. *Autonomous Control of an Unstable Helicopter Using Carrier Phase GPS Only*. PhD thesis, Stanford University, March 1995.
- [7] GAVRILETS, V. *Autonomous Aerobatic Maneuvering of Miniature Helicopters: Modeling and Control*. PhD thesis, School of Aeronautics and Astronautics, June 2003.
- [8] GAVRILETS, V., MARTINOS, I., METTLER, B., AND FERON, E. Control logic for automated aerobatic flight of miniature helicopter. In *AIAA Guidance, Navigation and Control Conference* (Monterey, CA, USA, August 2002), no. AIAA 2002-4834, AIAA.
- [9] HRABAR, S. E. *Vision-Based 3D Navigation for an Autonomous Helicopter*. PhD thesis, University of Southern California, Jan 2006.

- [10] JOHNSON, A., MONTGOMERY, J. F., AND MATTHIES, L. Vision guided landing of an autonomous helicopter in hazardous terrain. In *IEEE International Conference on Robotics and Automation* (May 2005), pp. 3977–3982.
- [11] JOHNSON, W. *Helicopter Theory*. Princeton University Press, Princeton, New Jersey, 1980.
- [12] KIM, B. S., AND CALISE, A. J. Nonlinear flight control using neural networks. *AIAA Journal of Guidance, Control, and Dynamics* 20, 1 (January-February 1997), 26–33.
- [13] KIM, N. *Improved Methods in Neural Network-Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, November 2003.
- [14] LEE, S. *Neural Network based Adaptive and its applications to Aerial Vehicles*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, April 2001.
- [15] MAHONY, R., AND HAMEL, T. Image-based visual servo control of aerial robotic systems using linear image features. *IEEE Transactions on Robotics* 21, 2 (April 2005), 227–239.
- [16] MEJIAS, L., ROBERTS, J., CAMPOY, P., USHER, K., AND CORKE, P. Two seconds to touchdown. vision-based controlled forced landing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing, China, October 2006), p. to appear.
- [17] MEJIAS, L., SARIPALLI, S., CAMPOY, P., AND SUKHATME, G. Visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *Proceedings of IEEE International Conference on Robotics and Automation* (Orlando, Florida, May 2006), pp. 2503–2508.
- [18] MEJIAS, L., SARIPALLI, S., CAMPOY, P., AND SUKHATME, G. Visual servoing for tracking features in urban areas using an autonomous helicopter. *Journal Of Field Robotics* 23, 3-4 (April 2006), 185–199.
- [19] MEJIAS, L., SARIPALLI, S., SUKHATME, G., AND CAMPOY, P. Detection and tracking of external features in a urban environment using an autonomous helicopter. In *Proceedings of IEEE International Conference on Robotics and Automation* (May 2005), pp. 3983–3988.
- [20] MERZ, T., DURANTI, S., AND CONTE, G. Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In *International Symposium on Experimental Robotics* (Singapore, June 2004).
- [21] MONTGOMERY, J. F. The use autonomous flying vehicle (afv) project: Year 2000 status. Tech. Rep. IRIS-00-390, Institute for Robotics and Intelligent Systems Technical Report, Los Angeles, CA 90089-0273, 2000.
- [22] NARDI, F. *Neural Network based Adaptive Algorithms for Nonlinear Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, November 2000.
- [23] PROUTY, R. W. *Helicopter Performance, Stability and Control*. Robert E. Kreiger Publishing, 1990.
- [24] SARIPALLI, S., MONTGOMERY, J. F., AND SUKHATME, G. S. Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation* 19, 3 (June 2003), 371–381.
- [25] SHAKERNIA, O., MA, Y., KOO, T. J., AND SASTRY, S. S. Landing an unmanned air vehicle: vision based motion estimation and nonlinear control. In *Asian Journal of Control* (September 1999), vol. 1, pp. 128–145.
- [26] SHIM, H. *Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*. PhD thesis, University of California, Berkeley, 2000.
- [27] SUGENO, M., WINSTON, H., HIRANO, I., AND S.KOTSU. Intelligent control of an unmanned helicopter based on fuzzy logic. In *American Helicopter Society 51st Annual forum Proceedings* (Houston, May 1995), pp. 791–803.
- [28] UAVRF. Georgia Institute of Technology. UAV Research Facility. <http://controls.ae.gatech.edu/uavrf/>, 2006.
- [29] YUE, A., AND POSTLETHWAITE, J. Improvement of helicopter handling qualities using hinf optimization. In *Control Theory and Applications, IEE Proceedings D* (May 1990), vol. 137(3), pp. 115–129.

ESTIMACIÓN DE VARIABLES DE ESTADO UTILIZANDO EL FILTRO DE PARTÍCULAS: COMPARACIÓN CON EL FILTRO DE KALMAN

Marcos del Toro Peral⁺
marcos.deltoro@diesia.uhu.es

Fernando Gómez Bravo⁺
fernando.gomez@diesia.uhu.es

Alberto MartinhoVale^{*}
alberto.vale@albatroz-eng.com

⁺Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática - Univ. de Huelva
Escuela Politécnica Superior, Ctra. Huelva - Palos de la Frontera, S/N.
21819 La Rábida - Palos de la Frontera HUELVA (ESPAÑA)

^{*}ALBATROZ – Engenharia, Investigação, Desenvolvimento e Inovação
Pólo Tecnológico de Lisboa, Centro de Incubação e Desenvolvimento
Estrada do Paço do Lumiar P-1600-546 LISBOA (PORTUGAL)

Resumen

Dentro de los métodos probabilísticos para la estimación de estados, el Filtro de Partículas es una técnica novedosa que está despertando un gran interés. Su éxito se debe a su versatilidad, pudiendo aplicarse a cualesquier tipo de sistemas (lineales y no lineales) y modelos de ruido. En este artículo se presenta una comparación entre los resultados obtenidos al aplicar el Filtro de Partículas con los obtenidos al aplicar el Filtro de Kalman.

Palabras Clave: Filtro de Partículas, Filtro de Kalman, estimación de estados.

1 INTRODUCCIÓN

La estimación de las variables de estado de un sistema representa un elemento fundamental a la hora de analizar el comportamiento del mismo o de establecer una estrategia de control adecuada. Sin embargo, con frecuencia el valor de dichas magnitudes no está accesible directamente y es necesario implementar un sistema que permita estimar sus valores a partir de los datos proporcionados por los sensores y el conocimiento de la señal de entrada. Existen numerosas técnicas que abordan el problema de la estimación según un enfoque determinista [14], sin embargo cuando el sistema, la señal de entrada, y/o los sensores se ven

afectados por un ruido estocástico relevante se hace necesario introducir técnicas probabilísticas en la estimación de estados.

Considérese un sistema físico representado por un modelo matemático, donde es posible medir las señales de entrada y las señales de salida. Por lo general, dichas medidas serán el único conocimiento que se tenga de manera directa del comportamiento del sistema.

Existen tres motivos básicos por los cuales las teorías de control deterministas, en muchos casos, no proporcionan suficientes medios para estimar el valor del estado [9]. En primer lugar, ningún modelo matemático de un sistema real es perfecto. Las características que se modelan son necesariamente aproximaciones. Por tanto, en todo modelo existe incertidumbre debido a dicha aproximación.

Un segundo problema es que los sistemas dinámicos no sólo son guiados por las señales de entrada teóricas, sino que también influyen en su comportamiento perturbaciones que no pueden ser controladas o modeladas “determinísticamente”. Esto representa la incertidumbre asociada al sistema. Incluso la señal de entrada aplicada no será exactamente la deseada, sino que también estará afectada por algún ruido.

Por último, los sensores tampoco proporcionan una medida exacta de las variables de salida ya que

introducen sus propias dinámicas, distorsiones y ruidos.

Existen numerosas técnicas probabilísticas para solventar en mayor o menor medida estos problemas. Seguramente, la técnica más conocida es el Filtro de Kalman (FK) [6, 7, 13] o el Filtro Extendido de Kalman (FEK) [10]. Desarrollos más recientes son los métodos basados en rejillas (*Grid-Based Method*) [1, 12] y el Filtro de Partículas (FP) [2, 4, 11], del que trata este artículo.

El FK proporciona un estimador óptimo del estado del sistema, pero sólo para sistemas lineales y modelos de ruido Gaussianos [6]. Para sistemas no lineales, se puede utilizar el FEK, que aproxima el sistema por su linealización de primer orden [10]. Este método puede ser poco preciso para sistemas altamente no lineales.

Por otro lado, el FP es una técnica de estimación probabilística que últimamente está suscitando gran atención [3, 5]. Uno de sus atractivos es que combina la potencia de los métodos de Montecarlo con la inferencia Bayesiana, sin aumentar desmedidamente el tiempo de cómputo respecto a los métodos anteriores. Además, no impone ninguna restricción respecto al tipo de sistema o los ruidos que le afectan, siendo perfectamente válido para sistemas no lineales afectados por ruidos no Gaussianos.

En este artículo se presentan los resultados de una serie de experimentos realizados sobre un sistema lineal de segundo orden afectado de ruidos Gaussianos. Se compara la estimación proporcionada por el FK con la generada aplicando el FP. Se ha decidido experimentar con este tipo de sistema ya que para él la estimación proporcionada por el FK es óptima. Es decir, se pretende comparar el FP con por el FK en la situación más favorable para este último. No obstante, es importante resaltar que la implementación del FP aplicada a estos experimentos sería la misma aunque el sistema fuese no lineal y los ruidos no Gaussianos.

El presente artículo se ha estructurado de la siguiente manera. En la sección 2 se exponen los principios básicos de la estimación de estados mediante técnicas probabilísticas. En la sección 3 se profundiza en la implementación del FP usada en las simulaciones. En la sección 4 se realiza una primera confrontación entre el FK y el FP, analizando de forma cualitativa las diferencias entre ambos y, por último, en la sección 5 se presentan los resultados de las simulaciones realizadas y las conclusiones obtenidas.

2 ESTIMACIÓN DE ESTADOS MEDIANTE TÉCNICAS PROBABILÍSTICAS

El problema de la estimación de estados mediante técnicas probabilísticas se puede formular de la siguiente manera [10]. Sea

$$\begin{aligned} X_{k+1} &= f(X_k, u_k, w_k) \\ Z_k &= h(X_k, v_k) \end{aligned} \quad (1)$$

el modelo de un sistema no lineal y variante en el tiempo, donde:

- k hace referencia al instante de tiempo
- X es el vector de estados.
- $f(\dots)$ define la dinámica del sistema.
- u es el vector de entradas.
- w es el vector que modela las fuentes de error del sistema.
- Z es el vector de las medidas proporcionadas por un sensor.
- $h(\dots)$ describe el modelo del sensor.
- v modela las fuentes de error en la medida.

Conocidos:

- f , h , la caracterización del ruido y las condiciones iniciales.
- La señal de entrada u
- La señal medida por el sensor Z

Se trata de obtener la mejor estimación posible de X_k con objeto de minimizar el error de la estimación según algún criterio.

Desde un punto de vista Bayesiano, el estimador debe propagar una función de densidad de probabilidad (fdp) del vector de estado (esto es, una función que define con qué probabilidad unos valores dados del vector de estado pueden ser los valores reales en un instante de tiempo k) condicionada por los datos accesibles hasta el momento. Expresado más formalmente la fdp queda determinada por la siguiente expresión (2):

$$P(X_k | Z_1 \dots Z_k, u_0 \dots u_{k-1}) \quad (2)$$

Ecuación que define la probabilidad de que el estado en el instante k sea X conocidas todas las medidas proporcionadas por el sensor hasta $t=k$, y las entradas al sistema hasta $t=k-1$. A partir de esta fdp se obtiene el estimador que optimiza cierto criterio, como pueden ser la media, la moda o la mediana [10].

3 EL FILTRO DE PARTÍCULAS

El principal objetivo del FP es estimar el estado de un sistema (lineal o no lineal) a medida que este evoluciona en el tiempo, estando el sistema sometido a una serie de perturbaciones que, en el caso más general, seguirán una distribución de probabilidad no Gaussiana [11].

En el ejemplo considerado en este artículo, se pretende estimar el estado X de un sistema gobernado por (3).

$$\begin{aligned} X_{k+1} &= AX_k + Bu_k + Gw_k \\ Z_{k+1} &= CX_k + v_{k+1} \end{aligned} \quad (3)$$

Estas ecuaciones son la particularización de (1) para un sistema lineal, el cual se considera completamente observable con ayuda de la medida Z proporcionada por un sistema sensorial. Las señales de ruido w y v se modelarán mediante variables aleatorias Gaussianas de media 0 y matrices de covarianza Q y R , respectivamente (4) y (5).

$$Q = \begin{bmatrix} s_{Qx1}^2 & 0 \\ 0 & s_{Qx2}^2 \end{bmatrix} \quad (4)$$

$$R = \begin{bmatrix} s_{Rx1}^2 & 0 \\ 0 & s_{Rx2}^2 \end{bmatrix} \quad (5)$$

Como ya se ha mencionado anteriormente, las *fdp* de estos ruidos no tienen que ser necesariamente Gaussianas para usar el FP, pero sí es necesario que lo sean en el caso del FK, de forma que se ha optado por este tipo de ruido para que las comparaciones entre ambos métodos tengan mayor sentido.

El FP es un algoritmo basado en una simulación de Montecarlo que proporciona un conjunto de M muestras (las partículas) con unos pesos asociados a las mismas, que indican la importancia de cada una a la hora de estimar el estado. Para realizar dicha estimación se simula la evolución de las partículas, obteniéndose M posibles estados del sistema. A continuación se utiliza la información dada por los sensores para ajustar los pesos de las partículas, de forma que cuánto más “lejos” esté una partícula de la medida del sensor menor será su peso. Finalmente se realiza una estimación del estado real del sistema en base al conjunto de partículas y sus pesos.

En la figura 1 se muestra una descripción global del uso del FP para la estimación de estados.

El fundamento del FP es el siguiente [11]. En un instante dado k , en el espacio de estados es posible definir una *fdp* que indica la probabilidad de que un estado dado sea el estado real. El FP construye una

representación basada en muestras de dicha *fdp* y utiliza la medida proporcionada por un sensor para disminuir la incertidumbre de la misma.

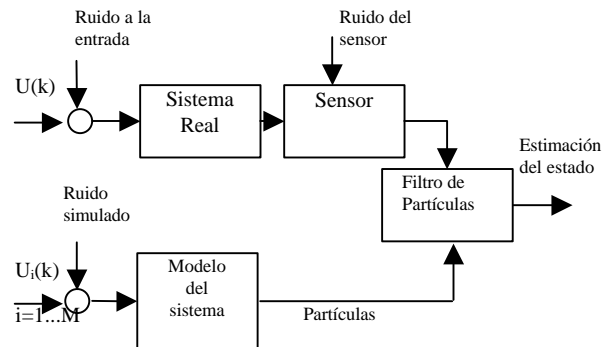


Figura 1: visión global del uso del FP

Para tal fin se utilizan las partículas, cada una con un peso asociado que denota la “calidad” de esa partícula, esto es, la probabilidad de que esa partícula represente al estado real. Mediante una media ponderada de estas partículas se obtiene una estimación del estado real del sistema.

El algoritmo del FP se puede desglosar en varias fases:

- Predicción
- Actualización
- Estimación
- *Resampling*

Las tres primeras se llevan a cabo en cada iteración del algoritmo, mientras que la tercera sólo cuando se cumplen ciertas condiciones, como se explicará en el apartado 3.4.

3.1 PREDICCIÓN

En esta fase del algoritmo, la evolución de las partículas se simula aplicando a cada una de ellas la entrada deseada más cierto ruido estocástico representado por una variable aleatoria (para simular la incertidumbre a la hora de aplicar la entrada del sistema), obteniéndose tantos estados posibles del sistema como partículas se usen. Este ruido a la entrada se considerará Gaussiano de media cero y varianza s_u y es independiente del ruido w de la ecuación (3).

Así pues, en cada iteración se calcula el nuevo estado de cada partícula según (6)

$$X_{k+1}^j = AX_k^j + B(U_k + N(0, s_u)) + N(0, Q) \quad (6)$$

donde el superíndice j hace referencia a la partícula.

3.2 ACTUALIZACIÓN

A continuación, el peso de las partículas es modificado en función de la información proporcionada por el sistema sensorial, de manera que las partículas que representen los estados más probables tendrán mayor peso.

Más concretamente, mediante simulación se calcula el valor que hubiese medido el sensor si el estado del sistema fuese realmente el dado por cada partícula. Así pues, esta simulación se realiza considerando la ecuación de salida de (3) a la que se le suma una variable aleatoria normal, resultando la ecuación (7)

$$Z_{k+1}^j = C \cdot X_{k+1}^j + N(0, R) \quad , j = 1 \dots M \quad (7)$$

Ahora bien, el peso de una partícula j será inversamente proporcional a la diferencia entre la medida del sensor para la partícula j (Z^j) y la medida real del sensor (Z). Esto es, cuanto más “lejos” esté la medida simulada de la partícula de la medida real del sensor, menor será su peso. Más formalmente, se ha tomado como peso de la partícula j (W_{k+1}^j) la probabilidad asociada a Z_{k+1}^j si el valor esperado de la medida fuese el proporcionado por los sensores. En el ejemplo considerado, ya que el modelo probabilístico del sensor es Gaussiano, se ha utilizado la siguiente expresión [11]:

$$W_{k+1}^j = P(X_{k+1}^j | Z) = \frac{1}{(\sqrt{2\pi})^2 S_{Rx1} S_{Rx2}} e^{-\frac{(z_1 - z_1^j)^2}{2S_{Rx1}^2}} e^{-\frac{(z_2 - z_2^j)^2}{2S_{Rx2}^2}} \quad (8)$$

donde $[z_1 \ z_2]^T$ sería la medida real del sensor en $k+1$, y $[z_1^j \ z_2^j]^T$ la medida que daría el sensor en $k+1$ si el estado fuese el dado por la partícula j .

3.3 ESTIMACIÓN DEL ESTADO

En esta tercera fase se realiza una media ponderada entre todas las partículas para obtener la estimación del estado real del sistema. Para esto se ha optado por realizar la media robusta de las partículas, ponderadas por sus pesos, esto es, no se hace la media de todas las partículas, sino sólo de aquellas cuyo peso difiera menos de un porcentaje dado del mayor peso.

$$X_{est} = \sum_j X^j w^j, j = 1 \dots M \quad / \quad w^j > k \cdot w_{max} \quad (9)$$

donde el índice j hace referencia a las partículas que cumplen $w^j > k \cdot w_{max}$, con k entre 0 y 1. Una vez seleccionadas las partículas que se van a utilizar para la media, se deben normalizar los pesos de las restantes. Es necesario resaltar que las partículas “no seleccionadas” sólo se descartan a la hora de realizar la estimación del estado, pero se siguen manteniendo para la iteración siguiente. Sólo en la fase de *resampling* se eliminan realmente las partículas con pesos despreciables y se sustituyen por copias de las partículas de mayor peso.

En [11] se sugieren otros métodos para llevar a cabo la estimación del estado, como la media ponderada o la mejor partícula.

3.4 RESAMPLING

Uno de los principales problemas que aparecen al usar el FP es la dispersión de las partículas después de unas pocas iteraciones, esto es, las partículas se van “alejando” del estado real más probable, por lo que aparecen muchas partículas con un peso ínfimo que prácticamente no contribuyen nada a la estimación del estado. Para evitar esto, cuando el número de partículas con peso despreciable supera cierto umbral, se lleva efectúa la fase de *resampling*, en la que se eliminan estas partículas con pesos demasiados pequeños y se realizan copias de las partículas con mayor peso para que la población no decrezca.

Para decidir cuándo realizar el *resampling* se ha usado el parámetro denominado “Tamaño Efectivo de las Muestras” ESS_t (*Effective Sample Size*), tal como se sugiere en [8], y que se calcula como se indica en (10).

$$ESS_t = \frac{M}{1 + cv_t^2} \quad (10)$$

donde cv_t^2 es el denominado “Coeficiente de Variación” (*Coefficient of Variation*) que viene dado por (11).

$$cv_t^2 = \frac{\text{var}(w_t(i))}{E^2(w_t(i))} = \frac{1}{M} \sum_{i=1}^M (M \cdot w(i) - 1)^2 \quad (11)$$

Este parámetro ESS_t da una idea del número de partículas con pesos no despreciables, de manera que cuando el ESS_t cae por debajo de cierto umbral (un porcentaje del número de partículas M) se produce el *resampling* de forma probabilística.

En [11] se proponen tres métodos para realizar el *resampling*, de los cuales se ha elegido el de “Selección con Reemplazamiento” (*Select with*

Replacement), también descrito en [2], según el cual una partícula tiene una probabilidad igual a su peso de seguir adelante (esto es, de no ser eliminada).

4 FILTRO DE PARTÍCULAS VS FILTRO DE KALMAN

Como es de sobra conocido, el FK es un conjunto de ecuaciones matemáticas que proporcionan un medio computacionalmente eficiente para estimar el estado de un sistema lineal sometido a diversos ruidos Gaussianos, de forma que se minimice el error cuadrático medio, a partir del modelo de estado del sistema y de la medida proporcionada por los sensores. Las ecuaciones que gobiernan el sistema son las mismas que en (3).

Bajo estas suposiciones para el ruido y para un sistema lineal, el FK proporciona una estimación óptima del estado [7,10]. El algoritmo del filtro se divide en dos etapas: predicción y corrección. En la primera se predice el estado del sistema en $t=k+1$ a partir de los datos (estado, entrada, matrices de covarianza) en $t=k$ y en la segunda etapa se corrige esta predicción introduciendo la medida del sensor en $t=k+1$. Las ecuaciones en (12) determinan el proceso de predicción y las de (13) el de corrección.

$$\begin{aligned}\hat{X}_{k+1}^* &= A\hat{X}_k + BU_k + GN(0, Q) \\ \hat{P}_{k+1}^* &= A\hat{P}_k A^T + GQG^T\end{aligned}\quad (12)$$

$$\begin{aligned}K_{k+1} &= \hat{P}_{k+1}^* C^T (C\hat{P}_{k+1}^* C^T + R)^{-1} \\ \hat{X}_{k+1} &= \hat{X}_{k+1}^* + K_{k+1} (Z_{k+1} - C\hat{X}_{k+1}^*) \\ \hat{P}_{k+1} &= (I - K_{k+1} C) \hat{P}_{k+1}^*\end{aligned}\quad (13)$$

Como se ha mencionado, el FK en sí, sólo se puede utilizar con sistemas lineales y siempre que la segunda ecuación de (3) que relaciona la medida del sensor con el estado también sea lineal. Para sistemas no lineales se debe usar el FEK que implica la “linealización” del sistema [6] donde la estimación ya no es óptima [10]. Sin embargo, el FP, como se ha señalado en los apartados anteriores, no requiere la linealidad ni del sistema ni de la ecuación del sensor, siendo, en cualquier caso, perfectamente válida la misma implementación del filtro ya descrita.

Por último, recordar que con el FK los ruidos del proceso y del sensor deben ser Gaussianos blancos y aditivos para poder aplicar las ecuaciones (12) y (13). Por contra, el FP no impone ninguna restricción a este respecto. Por supuesto, si se modelan los ruidos como no Gaussianos, la ecuación (8) que

calcula los pesos de las partículas debería sustituirse por otra expresión donde los pesos, presentarían una proporcionalidad inversa a la diferencia entre la medida real del sensor y la “medida” del sensor para cada partícula [4].

5 RESULTADOS DE SIMULACIÓN

Las simulaciones se han llevado a cabo sobre un sistema lineal de segundo orden subamortiguado, concretamente una masa m sujeta a un muelle y empujada sobre un plano horizontal con una fuerza F , como se muestra en la figura 3.

La ecuación que gobierna este sistema es (14). Si se reescribe en forma de variables de estado, donde x_1 será la posición x de la masa y x_2 la velocidad de la misma, se discretiza el sistema usando el método de Euler y se añaden las fuentes de ruido, se obtiene (15).

$$m\ddot{x} = F - m\dot{x} - kx \quad (14)$$

$$\begin{aligned}X_{k+1} &= \begin{bmatrix} 1 & \Delta t \\ -k\Delta t/m & 1 - m\Delta t/m \end{bmatrix} X_k + \\ &+ \begin{bmatrix} 0 \\ \Delta t/m \end{bmatrix} (F_k + N(0, S_u) + N(0, Q))\end{aligned}\quad (15)$$

$$Z_{k+1} = X_{k+1} + N(0, R)$$

En estas ecuaciones, Δt es el paso de integración.

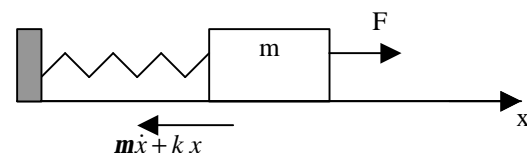


Figura 3: sistema a estudio

Las matrices G y C de (3) se han tomado como la matriz identidad.

En cuanto a los coeficientes de rozamiento y la constante elástica del muelle se ha tomado $\mu = 1$ Kg/s y $k = 0.2$ N/m, y la masa del objeto será $m = 100$ g.

El algoritmo del FP tiene varios parámetros que ajustar, como el número de partículas, la constante ‘ β ’ usada en el *resampling* y que determina cuándo es necesario hacer *resampling*, y la constante ‘ k ’ que determina qué porcentaje de las partículas se utilizan

en la estimación. En los resultados mostrados a continuación se ha tomado $\beta = 0.5$ y $k = 0.75$.

Por último se han usado las siguientes matrices de covarianza:

$$Q = \begin{bmatrix} 0.001^2 & 0 \\ 0 & 0.001^2 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.05^2 & 0 \\ 0 & 0.01^2 \end{bmatrix} \quad (17)$$

$$s_u = 0.01$$

El paso de integración en todas las simulaciones ha sido $\Delta t = 0.1s$

5.1 EXPERIMENTO 1: ENTRADA ESCALÓN

En este primer experimento se somete el sistema a una entrada escalón como se muestra en la figura 4 y se deja evolucionar hasta $t = 300s$.

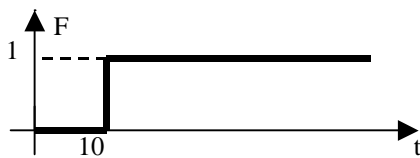


Figura 4: entrada del sistema

En la figura 5 se muestran simultáneamente tres gráficas: la evolución del estado real del sistema (negro y línea punteada), el estado estimado por el FP con 100 partículas (en azul y línea continua) y el estado medido por los sensores (verde y línea continua con puntos). En la figura 6 se comparan los errores absolutos cometidos en la posición y la velocidad usando la estimación del FP (en azul y línea continua) y utilizando sólo la información sensorial (en negro y con línea punteada).

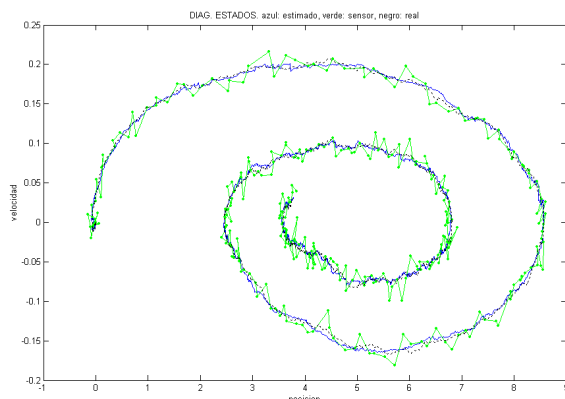


Figura 5: Diagrama de estados usando el FP (100 partículas)

Finalmente, en las figuras 7 y 8 se comparan los errores cometidos al usar el FK (en azul –más oscuro si B/N- y línea continua con puntos) y al usar el FP

(en rojo y línea continua). Cada figura representa un experimento con un número distinto de partículas, 50 en la figura 7 y 100 en la figura 8.

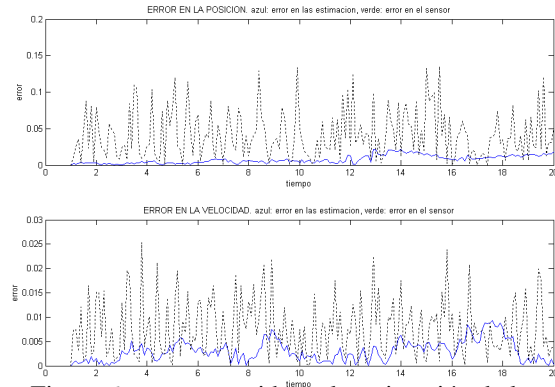


Figura 6: error cometido en la estimación de la posición (arriba) y en la velocidad (abajo) usando el FP (100 partículas)

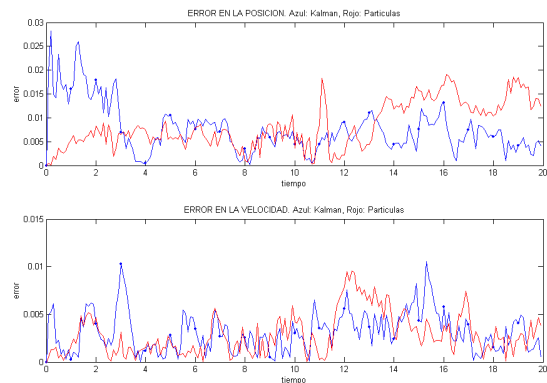


Figura 7: error cometido en la estimación de la posición (arriba) y en la velocidad (abajo) usando el FP (50 partículas) y el FK

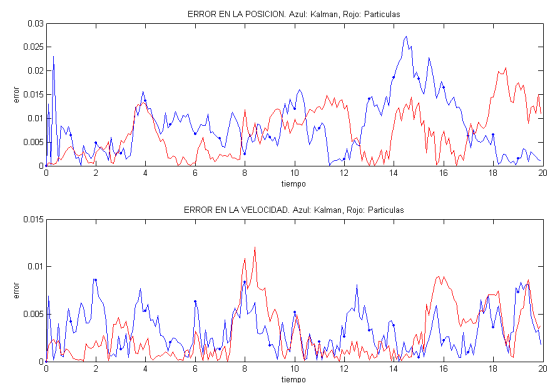


Figura 8: error cometido en la estimación de la posición (arriba) y en la velocidad (abajo) usando el FP (100 partículas) y el FK

Empíricamente se ha comprobado que con 100 partículas prácticamente se obtienen resultados equivalentes a los del FK y que para un mayor número de partículas los resultados obtenidos son similares.

Por último en la tabla 1 se puede ver la media de los errores medios cometidos al usar el FK y el FP y el tiempo medio de cómputo necesario para realizar 300 iteraciones en 50 experimentos. Esto es, se han realizado 50 experimentos y se ha calculado la media de los errores medios de cada experimento, así como la media del tiempo invertido en cada simulación.

Tabla 1: errores medios y tiempo de cómputo

	Kalman	Part. 50	Part. 100	Part. 400
Error Posición	0.01547	0.02654	0.01948	0.01621
Error Velocidad	0.00268	0.00278	0.00237	0.00222
Tiempo de cómputo	0.07126 s	0.18782 s	0.26688 s	0.78438 s

Decir, sobre todo a la hora de dar significado a los tiempos de cómputo, que los algoritmos han sido implementados en MATLAB 6.5 y las simulaciones se han ejecutado en un Pentium IV a 3 Ghz, con 384 MB de RAM.

Se aprecia en la tabla que con 100 partículas los errores usando el FK y partículas son ya de la misma magnitud. Por encima de 100 partículas no se aprecia una mejora significativa

5.2 EXPERIMENTO 2: PÉRDIDA DE LOS DATOS DEL SENSOR

En este segundo experimento, la entrada es la misma que en el experimento anterior, pero entre los instantes de tiempo $t=50s$ y $t=150s$ se supone que el sensor deja de dar datos válidos, no pudiéndose ejecutar la fase de corrección en el FK ni de actualización en el FP. En las figuras 9 y 10 se comparan los errores cometidos en la estimación usando el FK (en azul –más oscuro si B/N- con puntos) y el FP (en rojo). Los experimentos se han realizado con 50 y 100 partículas, figuras 9 y 10 respectivamente. En estas gráficas, se observa que para este experimento el comportamiento del FP es mejor que el del FK incluso para $M=50$

5.3 CONCLUSIONES

A la vista de los resultados mostrados arriba, se puede concluir que el FP proporciona resultados de calidad similar a los obtenidos por el FK, con la ventaja de que los modelos de ruido no tienen porqué ser Gaussianos. Por supuesto, su principal desventaja respecto al FK es el tiempo de cómputo, casi 4 veces mayor usando 100 partículas. Sin embargo recordar que los datos mostrados en la Tabla 1 muestran el tiempo transcurrido en 300 iteraciones, por lo que una iteración sólo tardaría 0'8896 ms, lo que podría ser suficientemente rápido para cualquier aplicación

típica de control en tiempo real. También hay que tener en cuenta que el algoritmo se ha implementado en MATLAB, por lo que es de esperar un comportamiento más rápido implementándolo en un lenguaje más eficiente como C.

En futuros desarrollos se abordará la tarea de confirmar, mediante experimentos reales, el buen comportamiento del FP observado en las simulaciones

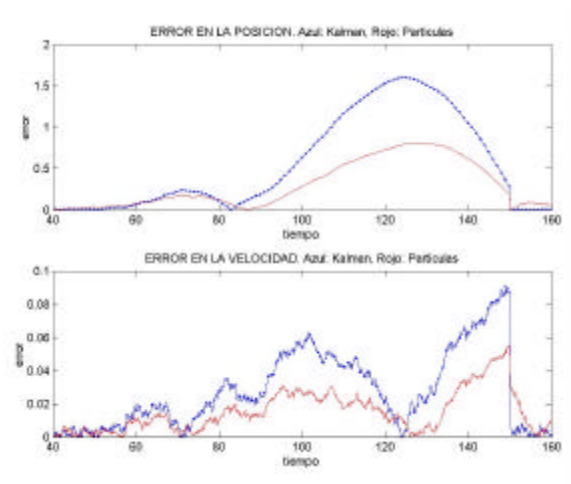


Figura 9: error cometido en la estimación de la posición (arriba) y en la velocidad (abajo) usando el FP (50 partículas) y el FK

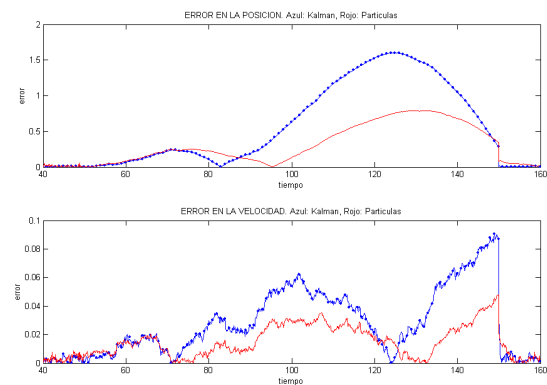


Figura 10: error cometido en la estimación de la posición (arriba) y en la velocidad (abajo) usando el FP (100 partículas) y el FK

Referencias

- [1] Burgard, W., Fox, D., Hennig, D., and Schmidt, T., (1996). "Estimating the absolute position of a mobile robot using position probability grids", en Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI-96), pp. 896–901.

- [2] Carpenter, J., Clifford, P. and Fearnhead, P. (1999) "*An improved particle filter for non-linear problems*". IEE proceedings – Radar, Sonar and Navigation, 146:2-7.
- [3] Doucet, A., de Freitas, N., and Gordon, N., "*Sequential Monte Carlo Methods in Practice*", Statistics for Engineering and Information Science, Springer, New York, 2001
- [4] Gomez Bravo, F., Vale A. and Ribeiro, M. I. (2006) "*Particle-filter approach and motion strategy for cooperative localization*", en Internacional Conference on Informatics in Control, Automation and Robotics 2006 (ICINCO - 2006)
- [5] Gordon, N. J., Salmond, D. J., and Smith, A. F. M., (1993) "*Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation*", IEE Proceedings-F, Vol. 140, No. 2, 1993, pp. 107-113
- [6] Grewal, M. S., and Andrews, A. P. (1993). "*Kalman Filtering Theory and Practice*". Upper Saddle River, NJ USA, Prentice Hall.
- [7] Kalman, R. E.. (1960) "*A new approach to linear filtering and prediction problems*". Trans. ASME, Journal of Basic Engineering, 82:35–45.
- [8] Liu, J. S., Chen, R. and Logvinenko, T. (2001) "*A theoretical framework for sequential importance sampling and resampling*". En A. Doucet, N. de Freitas and N.J. Gordon editors, Sequential Monte Carlo in Practice. Springer-Verlag, January.
- [9] Maybeck, Peter S. (1979). "*Stochastic Models, Estimation, and Control*", Volumen 1, Academic Press, Inc.
- [10] Ribeiro, M. I.. (2004) "*Kalman and Extended Kalman Filters: Concept, Derivation and Properties*", disponible en <http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf>
- [11] Rekleitis I. M. (2003) "*A Particle Filter Tutorial for Mobile Robot Localization*", en Internacional Conference on Robotics and Automation 2003 (ICRA-2003).
- [12] Schiele, B. and Crowley, J. L., (1994) "*A comparison of position estimation techniques using occupancy grids*" en Proceedings of IEEE Conference on Robotics and Automation (ICRA), vol. 2, pp. 628–1634.
- [13] Welch, G. and Bishop, G. (2001) "*An introduction to the kalman filter*". en Siggraph 2001 course material. Disponible en http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf
- [14] Åström, K. J., Wittenmark, B. (1988). "*Sistemas Controlados por Computador*". Ed. Paraninfo

SEGMENTACIÓN POR COLOR PARA EL SEGUIMIENTO DE OBJETOS MEDIANTE SNAKES.

Sandra Robla Gómez

Carlos Torre Ferrero, E.G. Sarabia, J.R. Llata

Departamento de Tecnología Electrónica e Ingeniería de Sistemas y Automática
E. T. S. de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria
Avda. Los Castros s/n, 39005, Santander, Cantabria, ESPAÑA.
E-mail: [srobla, carlos, esther, llata]@teisa.unican.es

Resumen

En este trabajo se presentan los resultados obtenidos en la realización de la segmentación por color de imágenes, empleando diferentes espacios de color y criterios de decisión, centrándose, en un primer lugar en la identificación de la piel para, posteriormente, realizar el seguimiento de una mano en movimiento. A continuación, para realizar dicho seguimiento, se ha aplicado la técnica de contornos activos estudiando a su vez su comportamiento en diversas condiciones. Como extensión a los resultados obtenidos se han aplicado ambas técnicas al seguimiento del movimiento de un brazo robot.

Palabras clave: Espacios de color, Segmentación de Imágenes, Clasificador Estadístico, Contornos Activos, Snakes

1. INTRODUCCIÓN

El seguimiento de objetos en una secuencia de imágenes es una de las tareas que más comunes dentro del campo de la visión artificial. Para desarrollar esta tarea existen diferentes técnicas como son imágenes en diferencia, correspondencia entre características, el cálculo de flujo óptico, o la aplicación de snakes, entre otras.

En este trabajo, se han aplicado contornos activos a la imagen resultante de la segmentación por color siendo ésta otra de las técnicas más empleadas dentro de la visión artificial. Existen diversos trabajos sobre la segmentación de la piel frente a fondos de imagen no uniformes. Partiendo de este hecho ha sido de nuestro interés realizar la segmentación de una mano frente a objetos de madera, para comprobar si poseen características de color muy similares como parece a priori. Por lo

que se ha realizado el estudio previo que se presenta a continuación.

2. ESPACIOS DE COLOR

El ojo humano percibe los colores en función de la longitud de onda que reflejan los objetos. Por lo que si las condiciones de iluminación varían también se modifica la forma en la que se percibe un mismo color. Sin embargo, el color se puede considerar como una de las características más importantes que definen a los objetos, lo que hace buscar soluciones a los problemas que plantean dichas variaciones de iluminación.

Como ya es conocido existen diversos espacios de color para la representación de imágenes, entre los cuales se encuentran el espacio RGB, el espacio HSV, o el espacio YCbCr.

2.1 ESPACIO RGB

Es el modo más utilizado en dispositivos de imagen que emiten luz, como televisiones o monitores. Este espacio de color presenta las imágenes como una combinación de los tres colores primarios rojo (R), verde (G), y azul (B).

$$X = R + G + B \quad (1)$$

donde x es el valor de color que puede tomar un píxel de la imagen.

Esta representación no hace distinción entre la luminosidad y la información del color, por lo que los algoritmos de segmentación de color realizados en este espacio no suelen ser demasiado robustos ante variaciones en las condiciones de iluminación.

2.2 ESPACIO HSV

El espacio de color HSV sí efectúa una separación entre luminosidad e información del color. Lo hace

definiendo por separado las componentes de tono, brillo y saturación.

El tono (H) se refiere a la longitud de onda reflejada o emitida por un objeto.

La saturación (S) es la intensidad del color.

El brillo (V), también llamando luminancia, se refiere a la claridad u oscuridad relativa a cada tono de color.

La transformación de RGB a HSV viene dada por las siguientes ecuaciones:

$$H = \arccos \frac{\frac{1}{2}((R-G) + (R-B))}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \quad (2)$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \quad (3)$$

$$V = \frac{1}{3}(R + G + B) \quad (4)$$

Las componentes H y S representan la cromaticidad, mientras que la componente V hace referencia a la luminosidad.

Este hecho, hace que los algoritmos de segmentación basados en este espacio de color no sean tan sensibles a variaciones de iluminación como lo pueden ser los basados en el espacio RGB.

2.3 ESPACIO YCbCr (YUV)

La transformación de RGB a YCbCr viene dada por las ecuaciones (5), (6), (7). La obtención de la luminosidad se realiza como la suma de los colores primarios asignando a cada uno un peso específico,

$$Y = 0.299R + 0.587G + 0.114B \quad (5)$$

y la obtención de la información del color se obtiene a través de las componentes C_b y C_r.

$$C_b = B - Y \quad (6)$$

$$C_r = R - Y \quad (7)$$

Este espacio de color es bastante utilizado en tareas de segmentación de color de piel ya que como reflejan las ecuaciones separa, al igual que el espacio HSV, las componentes de la luminosidad y la crominancia, haciendo que los algoritmos sean más robustos ante las variaciones de la iluminación.

2.4 ENSAYOS SOBRE IMÁGENES

Para hacer un pequeño estudio sobre el comportamiento de la piel, madera, y fondo en

estos espacios de color, se han tomado varias imágenes con pequeñas variaciones en la iluminación, en las que se ha seleccionado zonas de interés como son zonas de piel, zonas de madera, y zonas de fondo.

Las imágenes se han dividido en dos grupos, un primer grupo de entrenamiento a partir del cual se realizan todos los cálculos, y un segundo grupo de prueba. La Figura 1 muestra una de las imágenes del grupo de prueba.



Figura 1. Imagen capturada

Se han realizado las correspondientes transformaciones de color sobre dichas imágenes, y se ha realizado un análisis de la distribución de píxeles en los tres espacios de color, mostrando los resultados para color de piel, en color verde, color madera, en color amarillo, y fondo de las imágenes, en color azul.

Para el espacio RGB los resultados obtenidos son los siguientes:

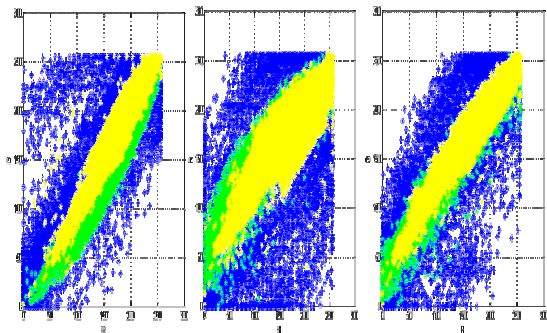


Figura 2. Distribución de píxeles en formato RGB

Se puede observar en la Figura 2 como los píxeles pertenecientes tanto a la piel como los pertenecientes a la madera se extienden de una forma lineal a lo largo del rango de valores RGB, encontrándose los grupos de píxeles de cada elemento muy solapados entre si.

Para el espacio HSV se obtuvieron los resultados que se muestran en la Figura 3. Se puede observar cómo la distribución de los píxeles tanto de la piel

como de la madera se produce a lo largo de todo el rango de valores de Tono (H), y cómo los diferentes grupos de píxeles se encuentran bastante solapados.

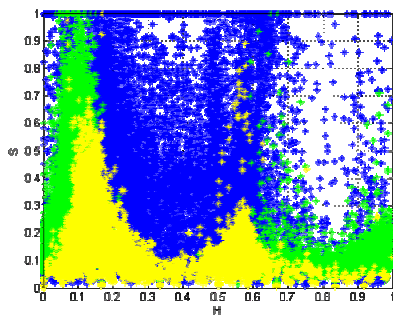


Figura 3. Distribución de píxeles en formato HSV

A continuación se muestran los resultados obtenidos para el espacio de color YC_bC_r

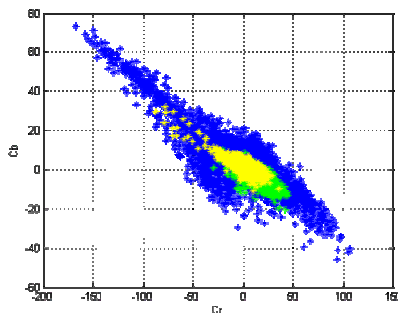


Figura 4. Distribución de píxeles en formato YC_bC_r

En la Figura 4 se puede ver cómo los píxeles tanto de piel como de madera se encuentran concentrados en una determinada zona del rango C_bC_r , existiendo entre ambas una cierta zona de solapamiento.

A la vista de los resultados para realizar una discriminación entre lo que es piel, madera y fondo se decidió generar funciones de distribución de probabilidad para cada uno de los estos objetos de estudio, teniendo en cuenta que el objetivo final es realizar la segmentación de la piel con el resto de la imagen, si bien era de esperar que para los espacios de color RGB y HSV no se iban a obtener resultados satisfactorios debido a la elevada desviación que presentan los píxeles tanto de piel como de madera a lo largo de sus respectivos rangos.

3. CLASIFICACIÓN

Para realizar la identificación de los píxeles que pueden pertenecer a la piel, madera, o fondo, se definieron tres clases: *clase piel*, *clase madera*,

clase fondo, y se crearon, a partir de una base de datos obtenida de varias imágenes de entrenamiento, las funciones de probabilidad correspondientes para cada una ellas con las que se puedan representar los valores de las características de cada clase, por ejemplo los valores de C_{rb} y C_r de la clase piel.

Estas funciones de densidad de probabilidad [2],[5] se crean a partir de la ecuación de distribución gaussiana (8)

$$p(c | piel) = \frac{1}{2\pi \sqrt{\sum_s}} e^{-\frac{1}{2}(c-\mu_s)^T \sum_s^{-1} (c-\mu_s)} \quad (8)$$

donde

c es el vector de características (C_b, C_r, H, \dots)

μ_s representa la media de los valores de las características, y aporta información sobre el punto donde se agrupan dichas características, ecuación (9)

Σ_s es la matriz de covarianza e indica cómo se agrupan las características entorno a la media, ecuación (10)

$$\mu_s = \frac{1}{n} \sum_{j=1}^n c_j \quad (9)$$

$$\Sigma_s = \frac{1}{n-1} \sum_{j=1}^n (c_j - \mu_s)(c_j - \mu_s)^T \quad (10)$$

Se han implementado las funciones de densidad de probabilidad para los espacios GB, RG, y, RB, existiendo en los tres casos un alto grado de solapamiento entre las tres clases, como por ejemplo se puede ver en la Figura 5 para el espacio GB.

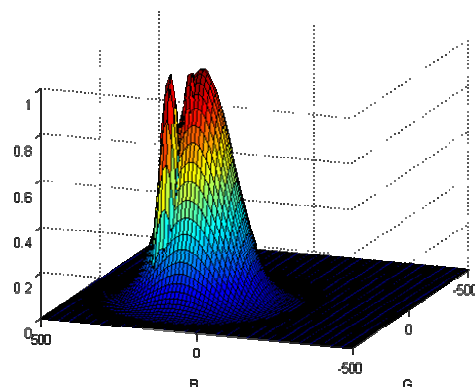


Figura 5. Función de densidad de probabilidad para clase piel, clase madera y clase fondo. (GB)

Para RG, y RB el resultado obtenido es similar al que se muestra en la figura por lo que se omite la presentación.

Para el formato HSV las funciones de densidad obtenidas igualmente presentan un alto grado de superposición entre las tres clases tal y como se aprecia en la representación tridimensional. Figura 6

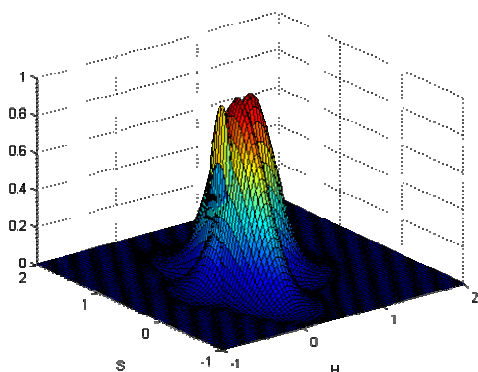


Figura 6. Función de densidad de probabilidad para clase piel, clase madera y clase fondo.(HS)

Si se analizan las gráficas obtenidas para el espacio de color YCbCr, la función de densidad de probabilidad obtenida para la clase piel se encuentra diferenciada del resto, presentando únicamente una pequeña zona de solapamiento con la clase fondo. Además se observa que la función de densidad correspondiente a la clase madera, está casi en su totalidad incluida en la clase fondo y diferenciada de la clase piel, Figura 7, y Figura 8.

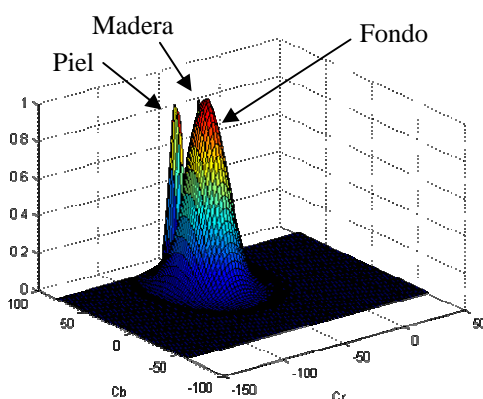


Figura 7. Función de densidad de probabilidad para clase piel, clase madera y clase fondo. $C_b C_r$

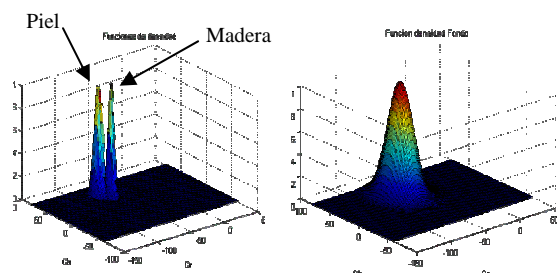


Figura 8. Funciones de densidad de probabilidad

A la vista de estos resultados se decidió incluir la clase madera dentro de la clase fondo, reduciendo las clases a *clase piel* y *clase no piel*. Por otra parte debido al alto grado de solapamiento entre las tres clases para los espacios RGB y HSV la aplicación de un clasificador bayesiano no iba a generar resultados satisfactorios en la segmentación, por lo que se decidió trabajar con el espacio de color $Y C_b C_r$.

Ya que las características del color no separan bien las clases definidas debido a su superposición en la distribución a lo largo de sus respectivos rangos, se ha utilizado un clasificador estadístico que emplea las distribuciones de probabilidad que caracterizan cada clase.

Según el teorema de Bayes [2]

$$p(\text{piel} | c) = \frac{p(c | \text{piel})p(\text{piel})}{p(c | \text{piel})p(\text{piel}) + p(c | \text{nopiel})p(\text{nopiel})} \quad (11)$$

donde :

$p(\text{piel} | c)$ representa la probabilidad a posteriori, es decir, la probabilidad de que un píxel de la clase piel posea las características c .

$p(c | \text{piel})$ representa la probabilidad de que un píxel de la imagen sea un píxel perteneciente a la clase piel.

$p(\text{piel})$ representa la probabilidad a priori de que un píxel de la imagen pertenezca a la clase piel

$p(\text{nopiel})$ análoga a $p(\text{piel})$

Se ha considerado que tanto $p(\text{piel})$ como $p(\text{nopiel})$ tienen la misma probabilidad de ocurrir, por lo que se les ha asignado a ambas el valor 0.5.

La regla de clasificación empleada es

$$\begin{aligned} \text{clase piel} & \quad \text{si } p(\text{piel} | c) > p(\text{nopiel} | c) \\ \text{clase no piel} & \quad \text{si } p(\text{nopiel} | c) > p(\text{piel} | c) \end{aligned}$$

En la Figura 9 se muestran dos de las imágenes que pertenecían del grupo de prueba y a las que se les aplicará el clasificador por color bayesiano.

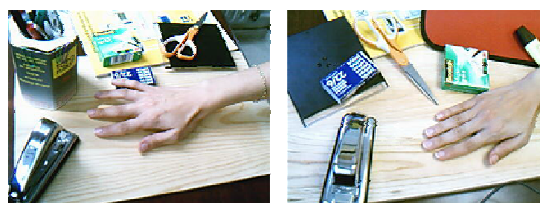


Figura 9. Imágenes de prueba

La Figura 10 presenta el resultado de la segmentación utilizando como criterio de decisión la regla de clasificación mencionada.

La figuras binarizadas corresponden al resultado de haber aplicado a la imagen segmentada las transformaciones morfológicas, de binarización, cierre y llenado de huecos. Figura 10

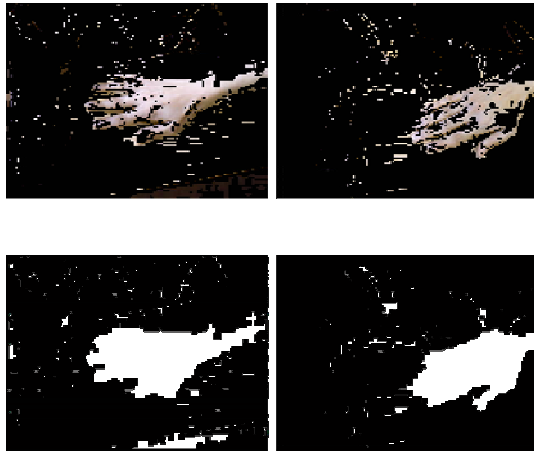


Figura 10. Imágenes segmentadas

Otro criterio de decisión fue el tomado a partir de las funciones de probabilidad calculadas mediante el clasificador bayesiano con las que se generó una imagen en escala de grises que representa los valores de $p(\text{piel}|\text{c})$. Para la obtención de las imágenes binarias se estableció un umbral que discriminara los valores más bajos de la probabilidad para la clase piel y se aplicaron las mismas operaciones morfológicas que en el caso anterior. Las imágenes de la parte superior de la Figura 11 muestran el resultado de la segmentación.

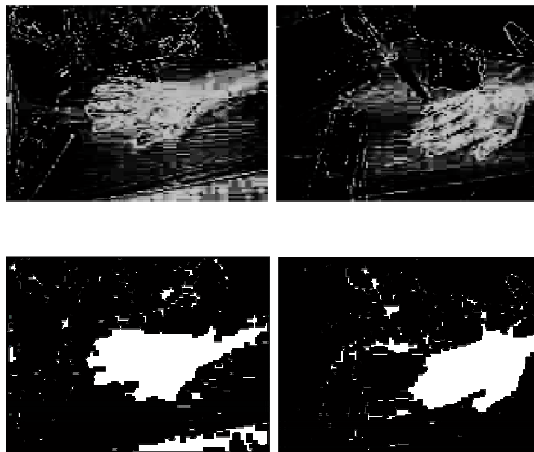


Figura 11. Imágenes segmentadas

Como se aprecia tanto en la Figura 10 como en la Figura 11, existen varias zonas en la imagen que han sido clasificadas como clase piel cuando en realidad pertenecen al fondo. Aún así, la segmentación se puede dar como aceptable, aunque susceptible de mejoras.

4. CREACIÓN DE SNAKES

Una vez realizada la segmentación por color para seguir el movimiento de la mano se generan contornos activos que se adaptan tanto al cambio de posición como de forma del objeto aplicado.

Un contorno activo se puede representar como una curva paramétrica $v(s,t)$, [1], [3], [4], dada por la ecuación (12)

$$v(s,t) = (x(s,t), y(s,t)) \forall s \Omega t \in T \quad (12)$$

donde s es el índice espacial definido para el intervalo Ω , y t es el tiempo definido en el intervalo T .

Esta curva, en la imagen, va modificando su forma hasta encontrar el mínimo de la función de energía [6], E_{snake} , definida en la ecuación (13)

$$E_{\text{snake}}(v) = \frac{1}{2} \int_{\Omega} [E_{\text{Int}}(v) + E_{\text{Ext}}(v)] ds \quad (13)$$

donde

E_{Int} , son las energías internas, que aportan restricciones de suavidad y están formadas por la suma de las energías elásticas, ecuación (14), y las energías de flexión, ecuaciones (15), (16) respectivamente

$$E_{\text{Int}} = E_{\text{Elastic}} + E_{\text{Bending}} \quad (14)$$

$$E_{\text{Elastic}}(v) = \alpha(s) \left[\frac{dv(s)}{ds} \right]^2, \quad v'(s) = \frac{dv(s)}{ds} \quad (15)$$

$$E_{\text{Bending}}(v) = \beta(s) \left[\frac{dv^2(s)}{ds} \right]^2, \quad v''(s) = \frac{dv^2(s)}{ds} \quad (16)$$

donde α , constante de tensión, y β constante de rigidez, dan peso a sus respectivas energías a lo largo de la curva.

E_{Ext} , son las denominadas energías externas, y hacen que la snake vaya hacia los límites buscados en la imagen. Vienen definidas por la ecuación (17)

$$E_{\text{Ext}}(v) = k \left[\nabla \left[G_{\sigma}(x,y) * I(x,y) \right] \right]^2 \quad (17)$$

donde ∇ es operador gradiente, k constante, $G_{\sigma}(x,y)$ es una función gaussiana con una desviación estándar σ , e $I(x,y)$ es una imagen en escala de grises. Si la imagen $I(x,y)$ está filtrada de

forma que contenga bordes, como va a ser el caso de este trabajo, la energía externa puede simplificarse y venir dada por

$$E_{Ext}(v) = kG_{\sigma}(x, y) * I(x, y) \quad (18)$$

El contorno activo que minimice la E_{snake} deberá cumplir la siguiente ecuación de Euler

$$\alpha v''(s) - \beta v'''(s) - \nabla E_{Ext} = 0 \quad (19)$$

es decir, tienen que igualarse las fuerzas internas y las externas. Para encontrar una solución a la ecuación (19), se realiza una discretización. Finalmente se tendrá un contorno $v(s)$ representado por una serie de puntos de control v_0, v_1, \dots, v_{n-1} , y una matriz K en la que estarán incluidas todas las constantes de peso de las energías, [5].

En la implementación de las snakes se ha incluido un algoritmo que reconoce si el objeto está o no dentro de la imagen, y en ese caso, en qué zona de la imagen se encuentra. Para ello la imagen se encuentra dividida en zona superior, inferior, derecha, e izquierda. Con esta información podremos calcular exactamente la orientación que debe tener la curva con la que se inicializa automáticamente la snake. Si el objeto se encuentra totalmente dentro de la imagen, la curva de inicialización se crea como una circunferencia centrada en el centro de masas del objeto, y con un radio proporcional a las dimensiones de dicho objeto.

Si parte del objeto está fuera de la imagen la curva de inicialización es la mitad de una elipse cuyo centro se sitúa en el centro del objeto y con radio mayor y menor obtenidos de las dimensiones de dicho objeto, con lo que se generará un contorno activo abierto. Después de una primera curva de inicialización el contorno activo sufre sucesivas transformaciones que hacen que el valor de la curva tienda al mínimo de energía buscado.

Para mantener el contorno activo abierto, en las ecuaciones que deforman la snake se ha introducido una discontinuidad entre dos de los puntos de control que lo forman, desligando de este modo las energías que relacionan ambos puntos.

En la Figura 12, (imágenes con diferentes iluminaciones y orientaciones), se muestran, tras haber realizado la segmentación por color, los contornos activos abiertos aplicados a la primera imagen perteneciente a una secuencia de imágenes, y su evolución a lo largo del tiempo.

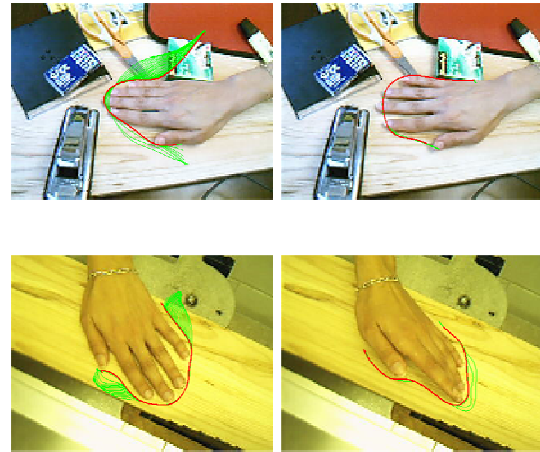


Figura 12. Snakes aplicadas a imágenes de prueba

5. SEGUIMIENTO DE UN BRAZO ROBOT EMPLEANDO SNAKES

Una aplicación que puede resultar interesante es realizar el seguimiento del movimiento de un brazo robot. En este caso se emplearon dos cámaras que captan imágenes de los desplazamientos del robot. La iluminación captada por una y otra cámara es diferente debido a su localización. Al igual que se realizó para el proceso de detección de una mano, se ha realizado la segmentación por color del robot en el espacio YC_bC_r , y se han aplicado sobre las imágenes segmentadas la técnica de las snakes.

La primera de las imágenes muestra cómo existen zonas donde el contorno activo no se ajusta perfectamente al objeto.

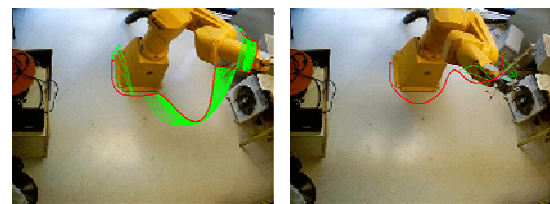


Figura 13. Contornos activos abiertos

Este problema suele desaparecer en las imágenes sucesivas, siempre y cuando el movimiento producido entre ellas no sea demasiado elevado, en cuyo caso la snake puede llegar a encontrar mínimos no deseados y por lo tanto no ajustarse de forma correcta a la imagen.

Las imágenes de la Figura 14 muestran los mismos movimientos del robot tomados desde otra cámara y su snake correspondiente. La inicialización para este caso se realiza por medio de un círculo (curva cerrada debido a que está completamente dentro la

imagen) situado en el centro del objeto y con un radio aproximado a sus dimensiones.

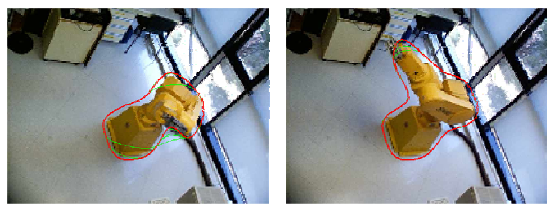


Figura 14. Contornos activos cerrados

Si se aplicara un contorno cerrado para el caso de una imagen cuyo objeto de estudio estuviera parte fuera de la imagen, el resultado no sería satisfactorio, ya que parte de la curva de inicialización de la snake queda fuera de la imagen, y al ser sometida al ajuste y deformación ya en la primera iteración perderá parte de los puntos que la forman haciendo como se puede observar en la Figura 15 que la adaptación al objeto se realice de forma incorrecta.

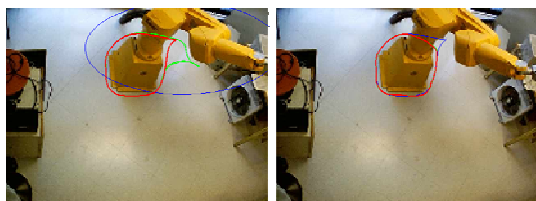


Figura 15. Problema de los contornos activos

Además si el objeto es excesivamente pequeño, la snake convergerá hacia mínimos no deseados haciendo que la snake quede atrapada en ellos, tal y como se puede apreciar en la Figura 16 donde además se muestra el mismo caso, pero solucionando este problema al aplicar un contorno activo abierto.

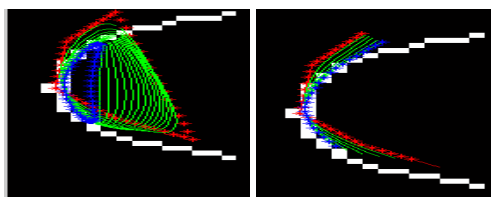


Figura 16. Problema de los contornos activos

6. CONCLUSIONES

Se han creado funciones de distribución de probabilidad en varios espacios de color, para objetos madera, color piel, y fondo, obteniendo los mejores resultados con el formato YC_bC_r . Se ha comprobado como la distribución de densidad de color de la piel humana, y de madera están

diferenciadas, y que las características que pertenecen a la madera se pueden englobar en características que definen un fondo.

Posteriormente se ha realizado la identificación de la mano empleando un clasificador clasificado estadístico de tipo bayesiano, obteniendo resultados satisfactorios, aunque existen pequeñas zonas segmentadas incorrectamente. Una posible mejora en la segmentación por color es el empleo de redes neuronales que una vez entrenadas realicen una mejor distinción entre los elementos de la imagen que sean piel y lo que no.

Se ha comprobado como el empleo de contornos activos es con la segmentación por color y clasificación bayesiana, una buena opción a la hora de realizar el seguimiento de objetos en movimiento, creando además contornos activos abiertos que permiten una mejor adaptación a objetos que no están completamente dentro de la imagen.

Las snakes se han aplicado además de para el caso del seguimiento de una mano, para el de un brazo robot. Ya que dicho movimiento se ha capturado a través de dos cámaras, el siguiente paso dentro de esta técnica será la realización de una superficie definida por snakes que proporcione información tridimensional acerca de dicho movimiento.

Referencias

- [1] Blake, A., (1998). Active Contours. Springer-Verlag.
- [2] De la Escalera (2001). Visión por Computador. Pearson Educación.
- [3] Lefèvre S., (2002), "An Extended Snake Model for Real-Time Multiple Object Tracking". International Workshop on Advanced Concepts for Intelligent Vision Systems. pp. 268-275
- [4] Leymarie, F.F. (1990), Tracking and describing deformable objects using active contour models. McGill University.
- [5] Vezhnevets V., (2003) "A Survey on Pixel-Based Skin Colour Detection Techniques". Graphicon-2003, pp.85-92
- [6] Xu, C. (1998). "Snakes, Shapes, and Gradient Vector Flow". IEEE Transactions on Image Processing, vol. 7, no 3, pp.359-359.

BARRIDO Y CONTROL DEL CICLO DE SERVICIO EN LA MODULACIÓN PWM MEDIANTE MICROCONTROLADOR: UNA APLICACIÓN A LOS GENERADORES FOTOVOLTAICOS

Andújar Márquez, José M.

Dpto. de Ing. Electrónica, de Sist. Infor. y Aut.- Universidad de Huelva.
Tlf: +34 959 217380, Fax: +34 959 217348, andujar@uhu.es

Durán Aranda, Eladio

Dpto. de Ing. Electrónica, de Sist. Infor. y Aut.- Universidad de Huelva.
Tlf: +34 959 217655, Fax: +34 959 217348, aranda@uhu.es

Resumen

La posibilidad de disponer de un sistema de control que permita realizar un barrido del ciclo de servicio en la modulación de ancho de pulso (PWM, Pulse Width Modulation), tanto en sentido ascendente (desde 0 hasta 1), como descendente (desde 1 hasta 0), así como la realización de barridos a intervalos discretos (0%-10%, 10%-20% ,.....,90%-100%), proporciona además de un control adicional sobre el convertidor, un método de evaluación de la respuesta de dicho convertidor, tanto en todo el rango del ciclo de servicio, como a intervalos. En este trabajo mostramos el diseño y la implementación de una tarjeta de ensayo, que permite realizar el barrido del ciclo de servicio, por tramos, y en todo el rango, así como para distintas frecuencias; esta tarjeta será utilizada para controlar un convertidor CC-CC, que a su vez permitirá la obtención de las curvas tensión-corriente ($I-V$) de un módulo fotovoltaico.

Palabras Clave: Barrido y Control del ciclo de servicio, Microcontrolador, convertidores CC-CC, módulos fotovoltaicos.

1 INTRODUCCIÓN

La técnica de modulación PWM es ampliamente utilizada en muy diferentes esquemas de control de convertidores de potencia, principalmente se basa en la modulación del ancho de un pulso (T_{ON}), mientras su periodo (T_S) se mantiene constante, esto permite que los dispositivos electrónicos que forman el convertidor actúen en conmutación, y por tanto con las mínimas pérdidas [1-2].

La relación entre el ancho de pulso y el periodo se le denomina ciclo de servicio (D), y esto permite establecer los dos parámetros característicos de una señal PWM: el ciclo de servicio y la frecuencia de

oscilación, que dará origen a la frecuencia de conmutación para los interruptores de potencia.

En el presente trabajo se presenta un método de realización del barrido de D (desde 0 hasta 1 y viceversa) empleando un microcontrolador, para distintas frecuencias y diferentes intervalos de duración del incremento y decremento de dicho ciclo de servicio.

Finalmente aplicaremos el barrido del ciclo de servicio obtenido para controlar un convertidor CC-CC, que a su vez reproducirá la curva $I-V$ de un módulo fotovoltaico.

2 BARRIDO DEL CICLO DE SERVICIO MEDIANTE μC .

El microcontrolador (μC) utilizado para implementar el barrido del ciclo de servicio es el PIC18F252 de Microchip, las características principales de este dispositivo pueden ser resumidas en las siguientes: emplea una arquitectura tipo RISC (Reduced Instruction Set Computer, arquitectura de procesador con juego reducido de instrucciones) para la unidad de control con un conjunto de 75 instrucciones, utiliza una memoria tipo FLASH (memoria programable no volátil de bajo consumo) de 32K de tamaño, más rápida y de mayor densidad que la memoria EEPROM, tolerando más escritura/borrado que ésta, de 256 bytes, la memoria de datos RAM tiene una capacidad de almacenamiento de 1536 bytes, dispone de 17 fuentes de interrupciones, tres puertos E/S, cuatro temporizadores internos, dos módulos Capture/Compare/PWM (CCP) que proporcionan señales moduladas en anchura de pulso, y un módulo de transmisión y recepción asíncrono y direccionable estándar (USART, Universal Serial Asynchronous Receiver Transmitter) es uno de los dos módulos de E/S serie. Este módulo es el que se usa para la comunicación serie con el PC.

Los dos módulos CCP, son denominados CCP1 y CCP2 (ambos módulos son prácticamente idénticos), contienen un registro asociado (CCPR1 y CCPR2) de 16 bits que puede trabajar como: Registro de 16 bits de Captura, Registro de 16 bits de Comparación, y Registro de Ciclo de Servicio del PWM.

La figura 1 muestra un diagrama de bloques simplificado del módulo CCP1 en modo PWM a nivel de registros. Los diagramas de flujo de los datos en el PIC cuando genera la señal PWM se muestran en la figura 2.

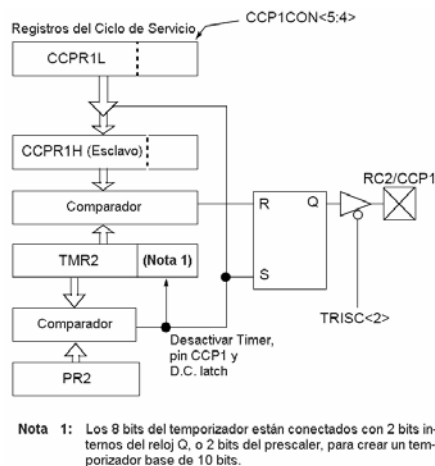


Figura 1: Diagrama de bloques simplificado del módulo CCP1 en modo PWM

El programa que gestiona el barrido del ciclo de servicio, ha sido desarrollado en lenguaje C, y modifica gradualmente (incrementa o decremente) la duración de T_{on} . Se ha dotado al sistema de la posibilidad de seleccionar mediante cinco pulsadores y un microinterruptor, un valor de frecuencia entre diez posibles, en el rango de 10KHz a 100KHz, a intervalos de una década.

Al ser pulsado uno de los pulsadores para establecer un valor de frecuencia concreto, se pone en marcha un mecanismo de tratamiento de la interrupción provocada que empieza con la identificación del pulsador accionado.

Mediante software también es posible establecer, intervalos de recorrido del ciclo de servicio entre el 0% y el 100%.

3 DESCRIPCIÓN DE LA APLICACIÓN.

3.1 Convertidores DC/DC como emuladores de resistencia variable.

Los convertidores DC/DC son muy empleados en aplicaciones donde se requiere obtener una tensión media de salida que puede ser mayor o menor que la aplicada a su entrada, gobernando los tiempos en que

el interruptor principal del convertidor conduce o no-conduce (técnica PWM), generalmente a frecuencia constante. A la relación entre el intervalo de tiempo en el que el interruptor está cerrado (T_{ON}) y el periodo de conmutación (T), se le denomina ciclo de servicio (D).

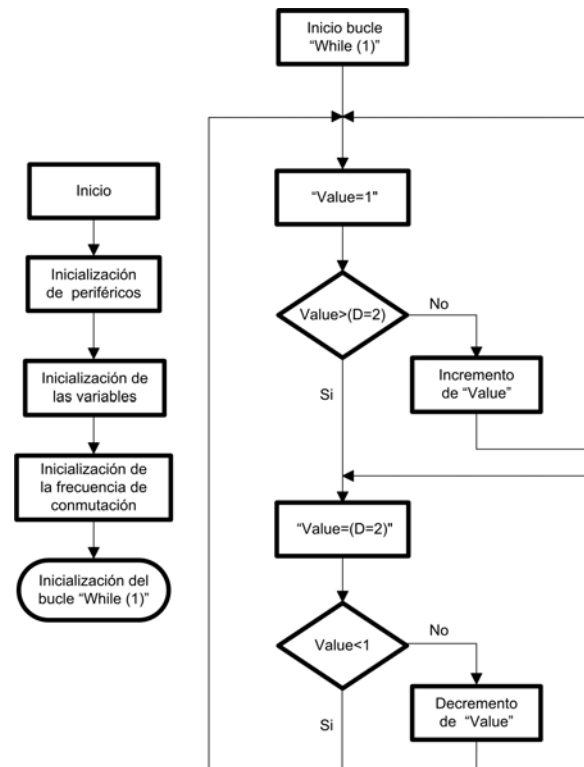


Figura 2: Diagramas de flujo del programa del PIC.

Tanto en el modo de funcionamiento con corriente continuada (CCM, *Continuous Conduction Mode*), como en el modo discontinuo (DCM, *Discontinuous Conduction Mode*), las diferentes topologías pueden ser asemejadas a un transformador de corriente continua [3-6], donde la relación de transformación puede ser controlada electrónicamente variando el ciclo de servicio D del convertidor en el rango $[0,1]$.

Suponiendo convertidores sin pérdidas, puede demostrarse [7-9] que las estructuras Reductor-Elevadoras son las únicas capaces de reproducir a su entrada una resistencia en el intervalo $[0,\infty)$. Una de las topologías que es capaz de reproducir todo el rango resistivo, es la topología SEPIC (*Single-ended primary inductance converter*), cuyo circuito eléctrico es mostrado en la figura 3.

3.2 Generador Fotovoltaico (FV).

La relación entre la tensión (V) y la corriente (I) proporcionada por un módulo fotovoltaico puede ser descrito mediante la expresión exponencial

simplificada propuesta por Gow y Manning [10], ecuación (1).

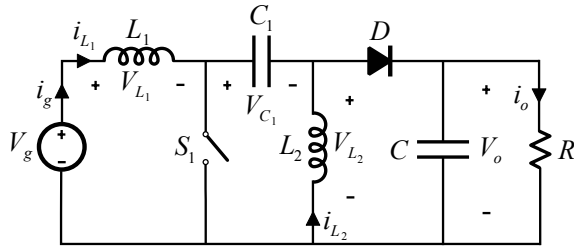


Figura 3: Circuito eléctrico de la estructura SEPIC.

$$I = n_p \left[I_L - I_s \left[e^{\frac{q \left(\frac{V}{n_s} + \frac{I R_s}{n_p} \right)}{A K T}} - 1 \right] - \frac{V/n_s + I R_s/n_p}{R_p} \right] \quad (1)$$

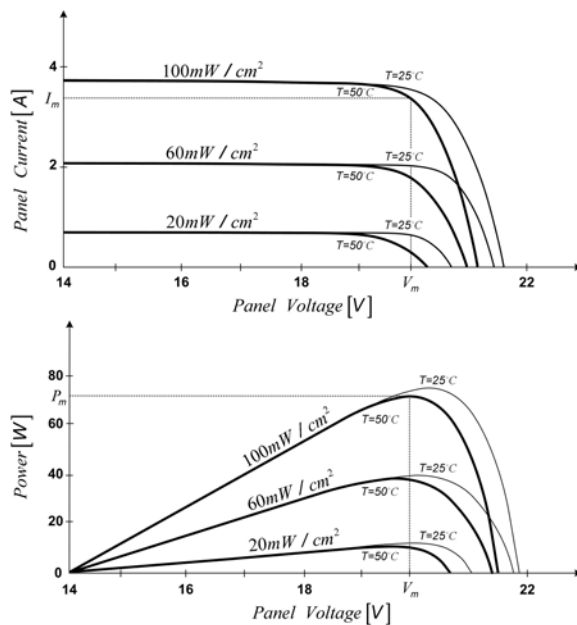


Figura 4: Características $I-V$ y $P-V$ de un panel fotovoltaico.

Donde n_p y n_s indican el número de células conectadas en paralelo y en serie; R_p y R_s las resistencias intrínsecas paralelo y serie asociadas a la célula, K es la constante de Boltzman ($1.38 \cdot 10^{-23}$ J/K) y q es la carga del electrón. El factor A determina la desviación de las características de una unión p-n ideal, e I_s es la corriente inversa de saturación, que presenta una dependencia con la temperatura del panel. I_L representa la corriente (foto corriente) generada por la radiación solar (S). Dicha corriente exhibe una relación sensiblemente lineal respecto de la radiación y la temperatura.

La expresión (1) (considerando la dependencia de sus

parámetros con T y S), proporciona las denominadas curvas $I-V$ teóricas de un panel fotovoltaico [11-14], y el producto de ambas magnitudes la potencia suministrada. En la Figura 4 se muestran las curvas de un panel determinado, así como la dependencia de estas variables con la temperatura y la radiación solar.

4 SISTEMA EXPERIMENTAL.

La estrategia descrita en la sección 2, ha sido implementada físicamente para obtener la curva $I-V$ de un módulo fotovoltaico de unos 100W, para lo cual se ha seguido el esquema mostrado en la figura 5.

En la figura 6, se muestra una fotografía de la placa de ensayo implementada, aquí se puede apreciar los cinco pulsadores (junto con sus respectivos circuitos antirrebotes) y el microinterruptor, que permiten seleccionar uno de los 10 valores discretos de frecuencia (en el rango 10KHz a 100KHz).

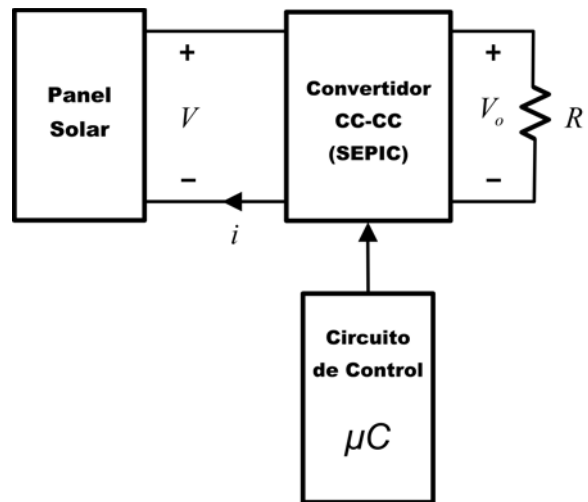


Figura 5: Diagrama de bloques del esquema experimental.

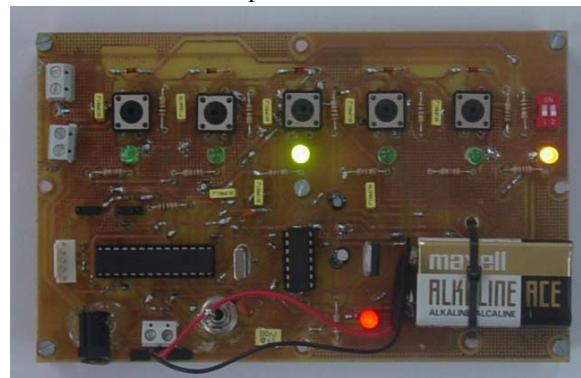


Figura 6: Placa de ensayo.

Se han añadido también señales ópticas, que junto con el correspondiente serigrafiado de la placa, permiten identificar rápidamente la frecuencia seleccionada.

En la figura 7, se muestra la placa donde se ha implementado el driver que permite accionar al Mosfet de potencia a partir de la señal generada por el microcontrolador, además del montaje del convertidor CC-CC en su versión SEPIC.

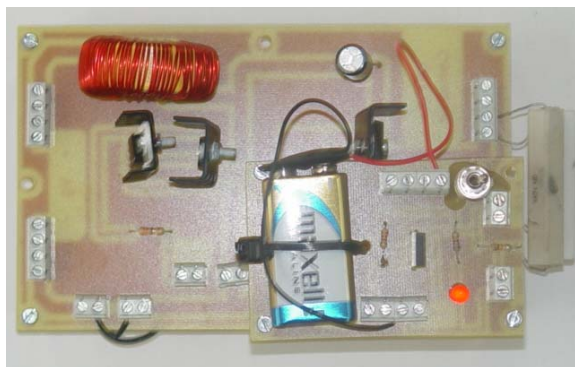


Figura 7: Driver de excitación y estructura SEPIC.



Figura 8: Diferentes ciclos de servicio y frecuencias medidos.

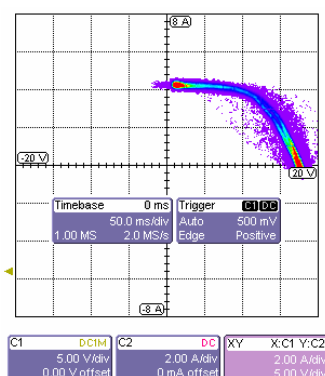


Figura 9: Curva I-V medida con el osciloscopio.

La figura 8 muestra tres señales de distintas frecuencias y ciclos de servicio generados por la placa de ensayo, y capturados por los tres canales de un osciloscopio digital. Así mismo, en la figura 9 aparece la curva $I-V$ capturada por el mismo osciloscopio en modo $X-Y$, donde el canal 1 (la tensión en los extremos del panel) es representado en el eje X y el canal 2 (la corriente suministrada por el módulo FV) es representada en el eje Y . Si bien la tensión en dicho panel ha sido medida directamente, la corriente proporcionada ha sido medida mediante una sonda basada en el efecto de Hall.

Para resaltar los puntos de la curva $I-V$ se ha ajustado la persistencia del osciloscopio hasta valores que permiten una representación clara.

5. CONCLUSIONES.

En el presente trabajo se ha descrito un método para realizar el barrido del ciclo de servicio de la modulación PWM mediante un microcontrolador, junto con sus diferentes opciones. El diseño de una placa de ensayo donde implementar la estrategia de control ha permitido evaluar la capacidad del convertidor SEPIC para reproducir la curva $I-V$ de un módulo fotovoltaico.

Referencias

- [1] Mohan, Undeland y Robbins "Power Electronics. Converters, Applications and Desing". John Wiley & Sons, 2002
- [2] Muhammad H.Rashid. "Power Electronics Handbook". Academia Press, 2001
- [3] K.K. Tse, M.T. Ho, H.S.-H. Chung, S.Y.R.Hui: "A comparative study of maximum-power-point tracker for photovoltaic panels using switching frequency modulation". IEEE Transactions on Power Electronics, vol. 17, no. 6, November 2002.
- [4] K.K. Tse, M.T. Ho, H.S.-H. Chung, S.Y.R.Hui: "A novel maximum power point tracker for PV panels using switching-frequency modulation scheme". IEEE Trans. on Industrial Elec., vol. 51, no. 2, April 2004.
- [5] S. Singer: "The application of loss-free resistors in power processing circuits". Power Elect., IEEE Trans., vol. 6, no. 4, pp. 595-600, 1991.
- [6] C. Jingquan, R. Erickson, D. Maksimovic: "Averaged switch modeling of boundary

- conduction mode DC-to-DC converters". Industrial Electronics Society, IECON '01. The 27th Annual Conference of the IEEE, vol. 2, pp. 844 – 849, 29 Nov.-2 Dec. 2001.*
- [7] J.M. Enrique, E. Durán, M. Sidrach-de-Cardona, J.M. Andújar, M.A. Bohórquez, J.E. Carretero, "A new approach to obtain I-V and P-V curves of PV panels by using DC-DC converters" Proc. of the 31st IEEE Photovoltaic Specialists Conference, PVSC 2005, Orlando, EEUU. pp. 1769-1772, January 3-7, 2005.
- [8] M.A. Bohórquez, J.M. Enrique, E. Durán, M. Sidrach-de-Cardona, J.E. Carretero, J.M. Andújar, "Analysis and monitoring of failures in PV panels by means of I-V and P-V curves using DC-DC converters", Proc. of the World Renewable Energy Congress, WREC 2005, Aberdeen, Scotland, UK. pp. 477-483, May 22-27, 2005.
- [9] E. Durán, J.M. Enrique, M.A. Bohórquez, M. Sidrach-de-Cardona, J.E. Carretero, J.M. Andújar, " A new Application of the Coupled-Inductors SEPIC Converter to obtain I-V and P-V Curves of Photovoltaic Modules", Proc. of the 11th European Conference on Power Electronics and Applications, EPE 2005, Dresden, Germany; Sep. 11-14, 2005.
- [10] J.A. Gow, C.D. Manning: *"Development of a photovoltaic array model for use in power-electronics simulation studies". IEE Proc. on Electric Power Applications, vol. 146, no.2, pp. 193-200, March 1999.*
- [11] CIEMAT: *"Fundamentos, dimensionado y aplicaciones de la energía solar fotovoltaica", Cap. 11, Editorial CIEMAT, Madrid 2000.*
- [12] D.L. King: *"Photovoltaic module and array performance characterization methods for all system operating conditions". Proceedings of NREL/SNL Photovoltaic Program Review Meeting, pp. 347-368, November 1996.*
- [13] C.M. Whitaker, T.U. Townsend, J.D. Newmiller, D.L. King, W.E. Boyson, J.A. Kratochvil, D.E. Collier, D.E. Osborn: *"Application and validation of a new PV performance characterization method". Photovoltaic Specialists Conference, Conference Record of the Twenty-Sixth IEEE, pp. 1253-1256, 29 Sept.-3 Oct. 1997.*
- [14] M. A. de Blas, J. L. Torres, E. Prieto, A. García: *"Selecting a suitable model for characterizing photovoltaic devices". Renewable Energy, vol. 25, no. 3, pp. 371-380, March 2002.*

Algoritmo de validación de datos de EEG basado en características de la FFT

Leandro Bueno

Carretera de Campo Real, km 0,200, Arganda del Rey, lbueno@iai.csic.es

José Luis Pons

Carretera de Campo Real, km 0,200, Arganda del Rey, jlpons@iai.csic.es

Resumen

En este trabajo se presenta el estado actual del trabajo realizado en el Instituto de Automática Industrial para el desarrollo de una interfaz de control basada en señales de electroencefalografía (EEG). Uno de los problemas de los sistemas de interfaz basados en EEG es la presencia de ruidos en la señal que enmascaran los cambios sutiles en las características de estas señales. El algoritmo aquí presentado explora la señal en busca de cambios en los parámetros de frecuencia que caracterizan las señales. Para el análisis de frecuencia se han probado dos aproximaciones distintas, el análisis con la transformada de Fourier y el procesamiento de las bandas de frecuencias con un banco de filtros pasa-banda digitales. La posibilidad de implementación de ambos algoritmos en tiempo real nos puede brindar una implementación del control online basado en BCI con datos fiables.

Palabras clave: BCI, FFT, EEG, Validación de datos.

1 Introducción

Una de las líneas de investigación del grupo de bioingeniería del Instituto de Automática Industrial es el desarrollo de ayudas técnicas a la discapacidad, como el proyecto MANUS [2] en el que se desarrolló una prótesis de mano multifuncional controlada por señales de electromiografía (EMG). Esta prótesis añadía a las prótesis comerciales actuales el movimiento de prono-supinación de la muñeca y la posibilidad de un mayor número de agarres por el movimiento del pulgar en dos planos diferentes.

Para la utilización de un dispositivo de ayuda a la discapacidad como MANUS, el usuario necesita el control voluntario de un grupo muscular, que en algunos casos no es posible. En estos casos se hace imperativo la utilización de otros canales de información, estos canales pueden ser de los más distintos como el movimiento de la cabeza, movimiento de los ojos, o las señales de EEG. Actualmente en el Instituto de Automática Industrial se está

desarrollando un sistema de control basado en las señales de EEG.

Los sistemas de control BCI, cuyo desarrollo se inició en los últimos años, están pensados inicialmente para el control de interfaces de comunicación para personas con niveles de discapacidad muy elevados, como los pacientes que sufren los estados más adelantados de patologías como la esclerosis múltiple, o los pacientes que sufren accidentes en la médula en niveles elevados de la columna vertebral.

La investigación en el campo de la interfaz directa entre cerebro y ordenador posee dos vertientes con enfoques muy diferenciados por el tipo de señal utilizada, de un lado los sistemas no-invasivos basados en EEG y los sistemas invasivos, basados en electrodos implantados en la corteza cerebral para la captura de la actividad cerebral.

A continuación se presenta las características de la señal EEG, las características de los principales tipos de ruidos que se puede encontrar en la señal EEG y se presenta una propuesta de algoritmo de validación de los datos de EEG para su posterior utilización en un sistema de control de una interfaz basada en EEG.

2 La señal de Electroencefalografía

La señal de Electroencefalografía (EEG) aparece en la superficie del cuero cabelludo proveniente de la interacción de la actividad eléctrica en distintos puntos del cerebro. Cada uno de estos puntos del cerebro genera una actividad eléctrica diferente, esta actividad se suma en el volumen del cerebro y se filtra por las sucesivas capas de distintas características eléctricas existentes entre el cerebro y la superficie de la piel. Este filtrado atenúa las altas frecuencias de las señales y además realiza una mezcla espacial de las señales, resultando en una señal EEG que refleja señales generadas por todo el volumen del cerebro. Por el filtrado espacial de los tejidos, se aprecia un nivel de actividad común en todas las canales[3].

El análisis clásico de la señal EEG se basa en separar la actividad registrada en distintas ban-

das de frecuencia, definidas entre 0.5 y 30 Hz, aproximadamente. Por las características de filtrado de los tejidos intermediarios entre el cerebro y los electrodos, frecuencias superiores a 40 Hz no poseen amplitud diferenciable del ruido de la señal.

2.1 Características de las bandas de la señal EEG

La señal de EEG se subdivide en 5 bandas principales con distintas características de frecuencia y que representan distintas condiciones específicas respecto a la actividad cerebral [1].

2.1.1 Alfa

La banda Alfa presenta frecuencias entre 8 y 13Hz, aproximadamente, está presente normalmente en adultos despiertos, pero no está relacionado al procesamiento de información. La actividad en esta banda aumenta con el cerrar de ojos y disminuye con el aumento de la actividad cerebral, como la ejecución de cálculos matemáticos o con la concentración en una determinada actividad.

2.1.2 Beta

La banda Beta presenta frecuencias entre los 13Hz y los 30Hz, está presente en adultos despiertos, dominando la señal en el caso de estar procesando información o resolviendo problemas analíticos. La actividad en esta banda representa, además, actividad no-sincronizada del cerebro.

2.1.3 Delta

La banda Delta presenta frecuencias entre 0.1 y 3Hz, caracteriza el sueño profundo y refleja el pensamiento inconsciente. Es el ritmo dominante en niños de hasta un año. Gran parte de los individuos diagnosticados con problemas de déficit de atención aumentan la actividad en delta al intentar concentrarse, al contrario de lo considerado normal, que es un descenso de esta actividad al concentrarse.

2.1.4 Theta

La banda Theta, con frecuencias entre 4 y 8Hz, se intensifica en los estados de meditación y oración. Es un ritmo anormal en adultos despiertos, pero es normal en niños de hasta 13 años. Refleja un estado entre el sueño y el estado de alerta.

2.1.5 Gama

La banda Gama, con frecuencias entre 36 y 44Hz, está presente en todo el cerebro. Una buena memoria está asociada a un ritmo bien regulado de

40Hz, también se atribuye los problemas de aprendizaje a una deficiencia en el ritmo de 40Hz.

De las bandas existentes en la señal, las que mejor representan la actividad consciente del cerebro son las bandas Alfa y Beta. Por este motivo se emplean estas bandas de frecuencia como características para identificar los estados de actividad del cerebro. En la Figura 1 se aprecian algunas de las ondas citadas, además se presenta también los picos característicos observados en cuadros de epilepsia.

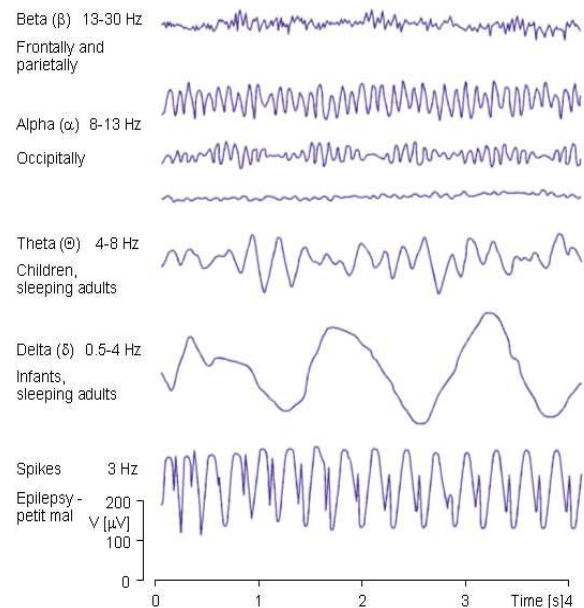


Figura 1: Componentes de las señales EEG. Imagen disponible en <http://butler.cc.tut.fi/malmivuo/bem/bembook/13/fi/1305.gif>

3 Ruidos en la señal EEG

Uno de los principales problemas para la implementación de los sistemas de control BCI es la presencia de ruido en la señal de electroencefalografía. Los principales tipos de ruidos que se puede identificar en la señal pueden ser de origen externo, como la inducción de tensiones a la frecuencia de la red eléctrica (50Hz), ruidos de EMG originados por la contracción de algún músculo cercano a los electrodos, desvíos de la señal originados por parpadeos o los generados por el movimiento de los ojos (Electro-oculografía, EOG). Cada uno de estos tipos de ruidos poseen características de frecuencia propias.

3.1 Electromiografía

La señal de EMG se caracteriza por presentar una respuesta de frecuencia con actividad más concen-

trada en bandas más elevadas que las bandas características de la señal EEG como se puede apreciar en la Figura 2. Por estas características de frecuencia, se puede eliminar gran parte de este ruido con un filtrado de la señal. La señal de EMG suele tener la amplitud mucho más grande que la señal EEG por existir menos tejido entre la fuente de la señal EMG, en este caso las unidades motoras, y los electrodos, localizados sobre la piel.

3.2 Ruidos inducidos por la red

El ruido inducido por la red se caracteriza por tener una frecuencia específica y muy bien caracterizada, 50Hz en el caso de Europa, y por algunas de sus armónicas, como se puede notar en la Figura 2f y 1h donde se aprecia una componente a 100Hz en el orden de las componentes de baja frecuencia características del EEG. Para estas situaciones un filtrado pasa-bajas también puede resultar eficiente y en los casos en los que el ruido a la frecuencia de la red aún se note en la señal, un filtrado rechaza-banda con una banda estrecha y centrado en la frecuencia de la red puede minimizar más todavía este ruido.

3.3 Ruidos de EOG y parpadeos

Los ruidos originados de los parpadeos suelen estar más concentrados en los electrodos más frontales, por estaren localizados más cerca de los ojos, pero en algunas situaciones también se puede apreciar este ruido en los canales más alejados de los ojos. El ruido de EOG se caracteriza por desvíos en la línea base de la señal EEG, estos desvíos pueden ser rápidos o lentos y en general se concentran en las posiciones más frontales.

Los parpadeos, a pesar de ser resultante de una actividad muscular no presenta la misma respuesta de la señal EMG ya descrita, sino que aparece como un desvío de la línea de base como los ruidos por EOG, pero los parpadeos son actividades puntuales, como se observa en la Figura 2g.

El procesamiento inicial de los datos es de extrema importancia para obtenerse una señal con un bajo nivel de ruido para que se pueda aplicar algunas técnicas de análisis no-lineal como la generación parámetros auto-regresivos que resultan ser sensibles a los niveles de ruido en la señal.

4 Análisis de frecuencia

Las señales EEG, por su característica de ser formada por la interacción de muchas componentes, se acerca mucho a una señal aleatoria pero, de acuerdo con la actividad mental del sujeto, la señal se torna más o menos correla-

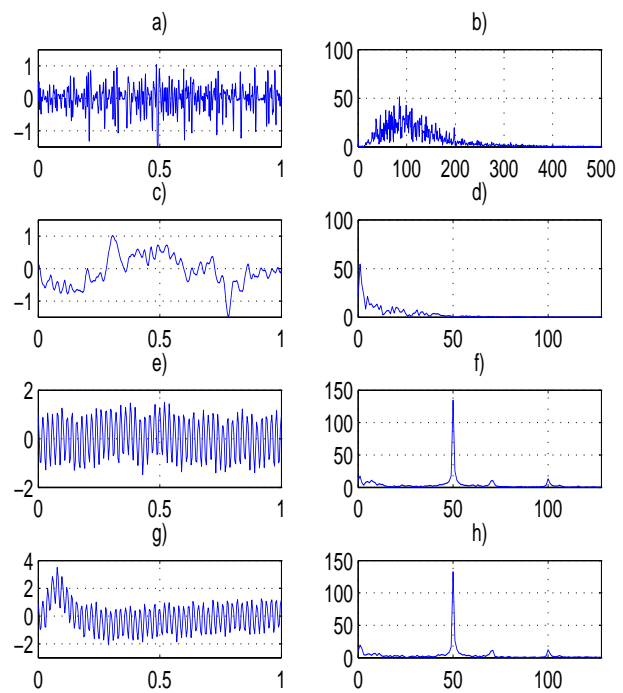


Figura 2: Señales EMG y EEG y sus contenidos de frecuencia: a) Señal EMG tomada en el bíceps; b) Información de frecuencia de la señal EMG; c) Señal DE EEG filtrada a partir de 40Hz, posición F4; d) Información de frecuencia de la señal EEG, posición F4; e) Señal de EEG contaminada por ruido de la red eléctrica (50Hz), posición F4; f) Información de frecuencia de la señal de EEG contaminada con ruido de la red, posición F4; g) Señal de EEG contaminada por la red y por un parpadeo, posición F4; h) Información de frecuencia de la señal contaminada por la red y por un parpadeo, posición F4

cionada. Esta correlación se evidencia por la sincronización/desincronización de algunas de las componentes de la señal. Este efecto se conoce como De/Sincronización Relacionada a Eventos (ERS/ERD). Este efecto se puede evaluar en el análisis de frecuencia por cambios en la distribución de las frecuencias.

Para el análisis de frecuencia de las señales EEG se pueden emplear distintas herramientas, como los bancos de filtros pasa-banda, la transformada de Fourier y el análisis basado en la transformada Wavelets. A continuación se describen algunas características de la implementación de un proceso de verificación de las señales para verificación de los niveles de ruido aleatorio en las señales EEG, basado en la FFT y en bancos de filtros.

4.1 La Transformada de Fourier

La transformada de Fourier es una herramienta clásica de procesamiento de señales que nos brinda

información del contenido de frecuencia en una determinada señal. En este trabajo se está utilizando la transformada de Fourier para caracterizar la señal en frecuencia, este análisis emplea la FFT para generar una distribución de frecuencias de cada uno de los canales adquiridos. El cálculo de la FFT se realiza a intervalos regulares y se utiliza para la caracterización el valor absoluto de las componentes que caracterizan las bandas *Alfa* y *Beta*.

4.2 Bancos de filtros pasa-banda

La implementación de los filtros pasa-banda, como en Tao et al. [4] genera a partir de las señales originales un conjunto de nuevas señales limitadas en banda. Las características de frecuencia en este caso son la energía de cada una de las bandas de cada uno de los canales. La estimativa de la energía de las bandas está representada por la envolvente de las señales filtradas. Partiendo de las características de la señal para dos grupos de tareas, imaginación de movimiento en el miembro superior derecho e izquierdo, el autor generó dos modelos de referencia, uno para cada actividad, y utilizó la correlación como medida de similitud entre el estado instantáneo y el modelo para clasificar la intención del sujeto. Para esta clasificación, el autor utilizó un subconjunto de los canales disponibles en los datos que maximizaban la información relevante a la tarea, aumentando la separación de las clases.

4.3 Aplicación a la verificación de los datos

La actividad eléctrica del cerebro cambia de acuerdo con el nivel de alerta o de actividad [3], una manera de verificar la calidad de los datos adquiridos es buscar por cambios en la actividad cerebral relacionados a la tarea. Partiendo de la premisa de que la actividad cerebral, en el caso de que el sujeto de pruebas está realizando una tarea específica, posee una determinada distribución de las características de frecuencia y en el momento en el que empieza una nueva tarea mental, esta distribución de las características debe cambiar a una nueva configuración, que caracterice esta nueva tarea.

Como medida de similitud de las distribuciones de frecuencia se utiliza la correlación, como en [4]. En este artículo el autor utiliza la correlación para identificar la similitud de las características de la señal adquirida con las características de las actividades de imaginación de movimiento de manos derecha e izquierda.

En este trabajo se utiliza la correlación bidimen-

sional definida como

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \bar{A})^2 \sum_m \sum_n (B_{mn} - \bar{B})^2}} \quad (1)$$

donde A y B son las matrices de características de las señales, de dimensiones $m \times n$, \bar{A} y \bar{B} están definidos como:

$$\bar{A} = \frac{\sum_m \sum_n A_{mn}}{m \cdot n} \quad (2)$$

Siguiendo esta premisa, se desarrolló un algoritmo de verificación a las señales adquiridas para validar estos para clasificación, configurándose así como una herramienta útil para las etapas iniciales del desarrollo de un sistema de control basado en señales de EEG.

El algoritmo de verificación calcula la correlación entre un conjunto de características y el calculado anteriormente en el tiempo, si esta correlación es siempre muy baja, se puede decir que la señal posee un nivel de aleatoriedad muy alto y que puede ser debido a elevados niveles de ruido en la banda de la señal de EEG. En los experimentos con datos validos se verifica una correlación elevada en el tiempo para una determinada actividad y una baja correlación en los cambios de actividad.

Si estos cambios en las estructura de las características no se verifican, o en todo el tiempo del experimento las variaciones son muy grandes, los niveles de ruido pueden estar demasiado elevados, y puede ser causado por una elevada impedancia entre los electrodos y el cuero cabelludo o una elevada impedancia en los electrodos de referencia del sistema de adquisición, necesitando un montaje más cuidadoso.

5 Comparaciones de los métodos de FFT y filtros pasa-banda

La FFT brinda la información de frecuencia en cada una de las ventanas temporales especificadas y la información de frecuencia en todas las bandas aparece en el mismo instante. El banco de filtros utiliza filtros pasa-bandas de diferentes frecuencias centrales, lo que genera retrasos de grupo en las señales filtradas. Estos retrasos de grupo son más evidentes en los filtros con frecuencias centrales más bajas, resultando en que la actividad representada en las ventanas de análisis de estas bandas de frecuencia estén desplazadas en el tiempo. Esta

característica genera problemas de sincronización entre las bandas indeseados.

La información de cada uno de los métodos de caracterización de frecuencia se organiza en una matriz que representa la información de cada canal en una determinada frecuencia, configurando una matriz $N \times N_f$, N es el número de canales y N_f es la cantidad de frecuencias de interés para el análisis. Para la identificación de niveles de ruido, se emplea toda la matriz de datos, en el caso de nuestro laboratorio, $N = 15$. Las frecuencias de interés, cubren las bandas *Alfa* y *Beta*. Para este estudio las frecuencias de interés para la FFT van de los 7 a los 24 Hz

Para el análisis con filtros de frecuencia, se utiliza un conjunto de filtros semejante al empleado en [4] que cubre un rango de frecuencias entre 5.25 y 30 Hz divididos en 6 bandas de anchos diferenciados de acuerdo con la frecuencia. Frecuencias más bajas poseen anchos de banda mas estrechos. Para la generación de las características con los bancos de filtros, se calcula una envolvente de la señal filtrada a través de la rectificación y filtrado de esta. El valor instantáneo de cada envolvente en cada canal de interés es una característica.

Los algoritmos presentados aquí pueden ser implementados en la plataforma software empleada en el laboratorio de adquisición de señales en tiempo real. Este estudio inicial se llevó a cabo sobre datos adquiridos de sujetos en nuestros laboratorios y sobre datos disponibles en internet para el BCI Competition III - Dataset IIIa (disponible en http://ida.first.fhg.de/projects/bci/competition_iii/).

A continuación se presentan algunos resultados del análisis basado en las características sacadas con la FFT de las señales del BCI Competition y de los datos sacados en experimentos realizados en nuestros laboratorios. Las características sacadas con los filtros no representaban las señales con la misma relación tiempo-frecuencia para todas las bandas de frecuencia, con la estructura propuesta, resultando en características no válidas, la investigación con otras estructuras de filtros con menor retraso de grupo puede resultar una alternativa a la FFT.

Las gráficas presentadas a continuación representan el promedio de las distintas evoluciones temporales de cada una de las tareas.

5.1 BCI Competition

Los datos de BCI Competition poseen tiempos de tareas diferentes de los tiempos de tarea de nuestro paradigma de experimento. En este paradigma, se presenta al sujeto una marca de inicio de tarea en el segundo 2 de la ejecución de la tarea y en

el segundo 5 esta marca se retira indicando al sujeto el inicio de la imaginación de movimiento, que debe seguir hasta el segundo 7. En la Figura 3 se aprecia la evolución de la correlación en el tiempo, para las dos tareas de interés y el promedio entre las tareas.

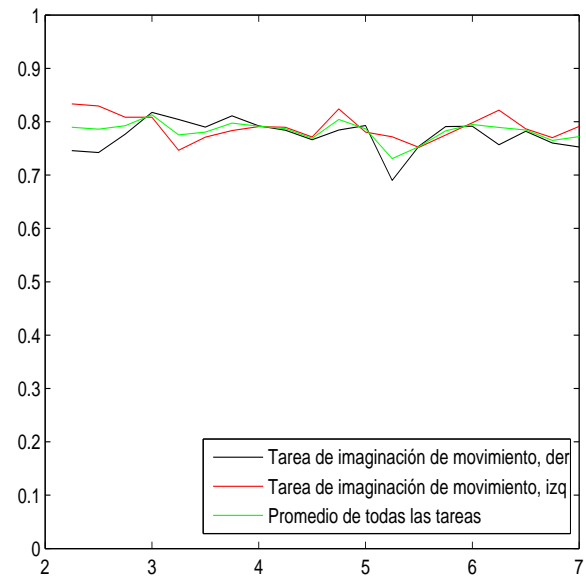


Figura 3: Evolución de la correlación temporal de las características. Datos de BCI Competition III.

Estas señales están libres de ruidos y se nota un descenso en la correlación después del quinto segundo, característico del cambio en la actividad mental. En una las tareas de este conjunto de datos, el cambio en la correlación es muy pequeño, mientras que en la otra tarea este cambio es bastante acentuado, representando incluso un mínimo en la correlación.

5.2 Datos Propios

El paradigma de los experimentos realizados en nuestros laboratorios posee algunas semejanzas con el ya presentado excepto por los tiempos de presentación de marcas. La marca de la tarea se presenta entre los instantes 5 y 6. La ejecución de la tarea empieza con la retirada de la marca de tarea.

Con uno de los sujetos hubieron problemas en el montaje de los electrodos. Estos problemas se caracterizan en el análisis con esta herramienta por una correlación baja y sin cambios representativos por la ejecución de la tarea, como se aprecia en la Figura 4. Los datos de esta sesión estaban contaminados por ruido de 50Hz y por ruidos en la banda de interés (entre 7 y 24Hz).

En la Figura 5 se puede apreciar la evolución de la correlación para unos datos válidos. En este caso los datos también contenían contaminación

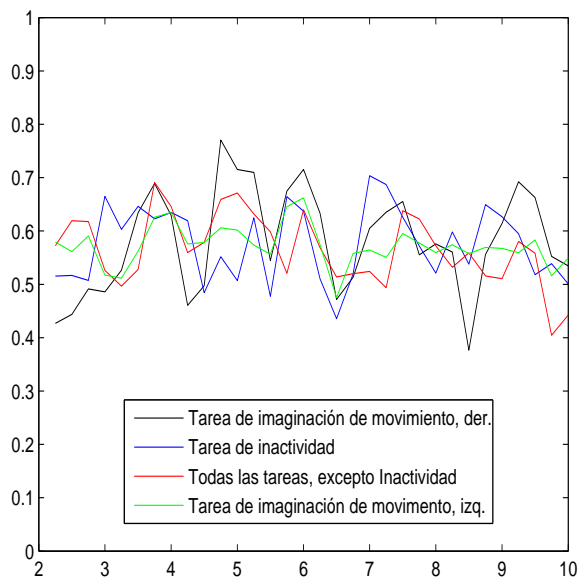


Figura 4: Evolución de la correlación temporal de las características. Datos propios, la baja correlación indica problemas en las señales

por ruido de 50Hz, pero los niveles de ruido en las bandas de interés no eran significativos para comprometer a los datos.

6 Conclusiones

La herramienta aquí presentada en su estado actual brinda la posibilidad de verificar la calidad de los datos en línea mientras se está ejecutando la sesión de medidas. La identificación de problemas de conexión es importante en el caso de no disponer de un medidor de impedancias en el laboratorio para verificación de las conexiones de los electrodos.

Como seguimiento de este trabajo, se están desarrollando los algoritmos necesarios para la implementación de la separación de las distintas tareas, basándose en las características de la señal. Otra línea de trabajo es la identificación de cambios en la distribución de las características con el aprendizaje.

Referencias

- [1] CrossRoads Institute, Brainwaves and EEG: The language of the brain, <http://www.crossroadsinstitute.org/eeg.html>
- [2] J.L. Pons, R. Ceres, (1998) MANUS: Modular Anthropomorphous User-adaptable Hand Prosthesis with Enhanced Mobility and Force Feedback, 3rd TIDE Congress on Technology for Inclusive Design and Equality, Helsinki.

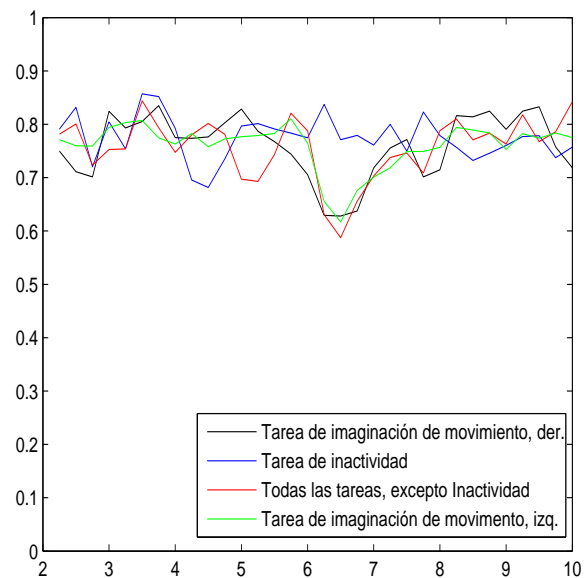


Figura 5: Evolución de la correlación temporal de las características. Datos propios, la alta correlación en la tarea, con disminución justo después de la transición entre tareas.

- [3] Jaakko Malmivuo, Robert Plonsey, (1995) Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields. Oxford University Press.
- [4] Tao Wang, Jie Deng and Bin He, (2004) Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns. Clinical Neurophysiology 115, 2744-2753.

APLICACIÓN DE MPC NO LINEAL BASADO EN MODELOS DE VOLTERRA A UNA PLANTA PILOTO

J. K. Gruber y C. Bordons

Dpto. Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingenieros
Universidad de Sevilla, Camino Descubrimientos, s/n., 41092 Sevilla
jgruber@cartuja.us.es, bordons@esi.us.es

Resumen

El presente documento muestra la aplicación de una estrategia de control predictivo no lineal basado en modelo (*NMPC*) a un proceso de laboratorio: una planta piloto con una válvula de recirculación manipulable para controlar la temperatura en el reactor. El controlador está basado en un modelo diagonal de *Volterra* de segundo orden para considerar efectos no lineales. Para calcular la acción de control se utiliza un método iterativo que requiere pocos recursos computacionales. El comportamiento del proceso y del controlador se presentan mediante resultados experimentales

Palabras clave: control predictivo no lineal, modelos de Volterra, secuencia pseudo-aleatoria.

1 INTRODUCCIÓN

Los modelos de *Volterra* representan la extensión simple y lógica de los modelos de convolución que han sido aplicados con gran éxito en control predictivo lineal basado en modelos (linear model predictive control, *MPC*). Estos modelos presentan genéricamente un buen comportamiento y se puede aprovechar su estructura en el diseño de controladores, especialmente en el caso de modelos de segundo orden. En el caso de un modelo de segundo orden y truncamiento de términos, el modelo tiene la forma

$$y(k) = h_0 + \sum_{i=1}^{N_1} a(i) u(k-i) + \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} b(i,j) u(k-i) u(k-j) \quad (1)$$

que corresponde al modelo lineal de convolución con la no linealidad como término adicional y aditivo.

El control predictivo no lineal basado en modelos (nonlinear model predictive control, *NMPC*) es un campo joven y todavía existen pocas referencias de

aplicaciones industriales [1]. Pero la capacidad de considerar dinámicas más complejas y no lineales convierte el *NMPC* en una opción prometedora en el momento de controlar procesos reales. Aunque los procesos industriales demuestran en general un comportamiento no lineal, la mayoría de los controladores predictivos están basados en modelos lineales. Sin embargo, los procesos fuertemente no lineales justifican el uso de *NMPC*.

En los procesos fuertemente no lineales la aplicación de *MPC* lineal puede ser poco efectiva y requiere el empleo de controladores no lineales. La principal ventaja de *NMPC* representa la posibilidad de considerar la dinámica no lineal. Enfrente a esta gran ventaja existen también desventajas que son: la obtención de un modelo no lineal a partir de datos experimentales y el problema de optimización del cálculo de la acción de control.

Este documento presenta la aplicación de control predictivo no lineal a un proceso de laboratorio. El modelo dinámico está basado en un modelo de *Volterra* y la optimización del cálculo de la acción de control se realiza mediante un método iterativo presentado por Doyle *et al.* [5]. El documento está distribuido de la siguiente manera: en el apartado 2 se explica el proceso utilizado y el problema que resulta del mismo, en el apartado 3 se presenta la identificación del modelo de *Volterra* y en el 4 se muestra la ley de control predictivo no lineal. En el apartado 5 se presentan los resultados de la aplicación del controlador no lineal al proceso real y en el 6 se presentan las conclusiones derivadas de la aplicación de este tipo de controlador.

2 DESCRIPCIÓN DEL PROBLEMA

Para la aplicación de un control no lineal se ha elegido un proceso real representado por una planta piloto, estudiado anteriormente por varios autores [6, 10, 2]. La planta piloto puede simular reacciones químicas exotérmicas a base de cambios en la temperatura. Los cuatro elementos principales de la planta son: el reactor, la resistencia, el intercambiador de calor y la válvula de recirculación, ver la figura 1.

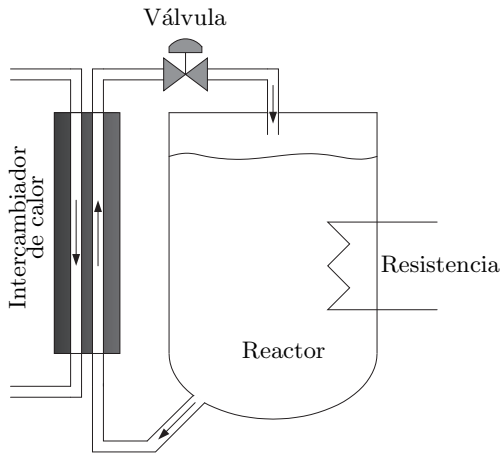


Figura 1: Diagrama de la planta piloto con los cuatro elementos: reactor, válvula de recirculación, resistencia y intercambiador de calor.

Mediante la resistencia se puede suministrar energía calórica y en consecuencia aumentar la temperatura del contenido del reactor. Para la reducción de la temperatura del contenido del reactor se utiliza el intercambiador de calor. El transporte de energía calórica desde el reactor al intercambiador de calor se puede regular mediante la válvula. El contenido saliente del reactor que pasa por el intercambiador de calor entra otra vez en el reactor tal que la planta piloto representa un sistema cerrado con respecto a la masa.

La planta piloto representada (ver la figura 1) se utiliza en general para simular experimentos de reacciones químicas exotérmicas. Para información más detallada sobre el proceso utilizado ver [2, 3]. En este documento no se utiliza un modelo matemático para calcular la energía calórica generada por una presunta reacción química, sino que se fija la resistencia a un valor constante. Con la resistencia fijada a un valor constante, eso significa que la resistencia no representa una entrada manipulable del proceso, la planta piloto representa un sistema *SISO* (single in single out) con una entrada (la apertura de la válvula) y una salida (la temperatura en el reactor).

La idea de este artículo es la obtención de un modelo matemático discreto para representar el cambio de la temperatura T en dependencia de la apertura de la válvula de recirculación v_8 con un suministro constante de energía (resistencia fijado a un valor constante) y su posterior uso para el desarrollo de un controlador predictivo no lineal. Tal modelo se puede escribir de la siguiente manera:

$$y(k+1) = f(u(k), u(k-1), u(k-2), \dots) \quad (2)$$

con la salida y (temperatura T) y la entrada u (apertura de la válvula v_8).

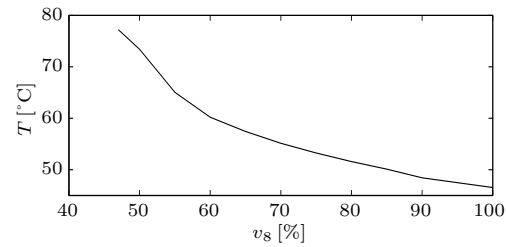


Figura 2: Relación entre apertura de la válvula y temperatura en régimen estacionario y resistencia con un valor constante del 50 %.

El modelo para identificar debería considerar la dinámica y los efectos no lineales del proceso de laboratorio. Una de las no linealidades es la relación entre la apertura de la válvula v_8 y la temperatura T en régimen estacionario y la resistencia con un valor constante, ver la figura 2.

3 IDENTIFICACIÓN DE MODELOS

La identificación de parámetros de modelos consiste en la mayoría de los casos de cuatro partes: elección de la estructura del modelo, elección de señales de entrada adecuadas, ensayos con aplicación de las señales elegidas para conseguir datos de tipo entrada-salida, así como la propia identificación de los parámetros del modelo.

3.1 ESTRUCTURA DE MODELO

A pesar del hecho que los sistemas físicos evolucionan en general en tiempo continuo, la mayoría de esquemas de control basados en modelos utilizan ordenadores para recoger las medidas de los sensores en el instante de tiempo discreto t_k y realizan las acciones de control algunos instantes más tarde en t_{k+n} . Por estas razones existe la necesidad de describir sistemas continuos mediante modelos en tiempo discreto. Los modelos de *Volterra* permiten la representación de sistemas complejos y no lineales mediante modelos discretos y por lo tanto su uso en el control de estos sistemas.

Finalmente, se ha elegido un modelo de *Volterra* de segundo orden que representa un modelo no lineal, con el fin de considerar la dinámica compleja del proceso real. La estructura general de un modelo de *Volterra* de segundo orden es la siguiente:

$$y(k) = h_0 + \sum_{i=1}^{\infty} a(i) u(k-i) + \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} b(i,j) u(k-i) u(k-j) \quad (3)$$

Con el fin de reducir considerablemente el número de parámetros para identificar, se ha elegido un modelo diagonal de *Volterra* que representa un caso especial del modelo de *Volterra*. En este tipo de modelo se consideran solamente los términos cuadráticos de los pasados valores de la entrada y no se utilizan los términos cruzados de los mismos. Esta reducción de parámetros resulta en la siguiente estructura de modelo:

$$y(k) = h_0 + \sum_{i=1}^{\infty} a(i) u(k-i) + \sum_{i=1}^{\infty} b(i) u(k-i)^2 \quad (4)$$

Con el truncamiento a N_1 términos lineales y N_2 términos no lineales el modelo diagonal de *Volterra* tiene la siguiente forma:

$$y(k) = h_0 + \sum_{i=1}^{N_1} a(i) u(k-i) + \sum_{i=1}^{N_2} b(i) u(k-i)^2 \quad (5)$$

3.2 SEÑAL DE ENTRADA

Para la identificación de los parámetros del modelo diagonal de *Volterra* hay que utilizar datos de tipo entrada-salida adecuados. La obtención de estos datos requiere la elección de una señal de entrada adecuada. La señal de entrada debe permitir la identificación de la dinámica compleja del sistema.

En muchas identificaciones de sistemas se utiliza como señal de entrada el ruido blanco de *Gauss* [8]. No obstante, el ruido blanco de *Gauss* es bastante atractivo en teoría, pero poco atractivo en el momento de aplicarlo al sistema [9]. El uso de ruido blanco como señal de entrada para una válvula resulta en un movimiento constante de la misma y puede dañarla. Por estas razones hay que encontrar una señal de entrada aplicable que no dañe la válvula pero permita la identificación de los parámetros del modelo.

Para la identificación de sistemas lineales se usa en muchos casos una *PRBS* (Pseudo Random Binary Sequences). Desafortunadamente, las *PRBS* tienen una gran deficiencia en el momento de identificar sistemas no lineales debido a la excitación insuficiente del sistema [8].

Por los problemas con el ruido blanco de *Gauss* y la *PRBS* se ha elegido finalmente como señal de entrada una *PRMS* (Pseudo Random Multilevel Sequences). Las *PRMS* tienen, al contrario que las *PRBS*, no solamente dos niveles, sino varios

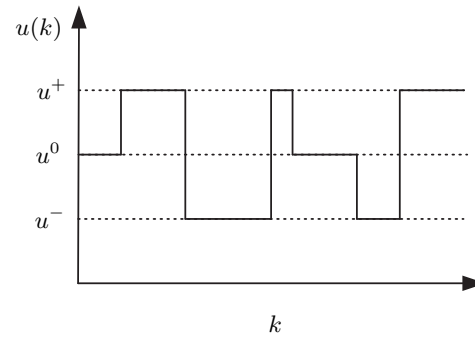


Figura 3: Secuencia (*PRMS*) de entrada para la identificación de modelos de *Volterra* hasta segundo orden.

niveles. La ampliación del número de diferentes niveles en la señal de entrada permite una mejor excitación del sistema. Las *PRMS* con $N + 1$ niveles pueden excitar suficientemente sistemas de *Volterra* de orden N [8]. Por el orden $N = 2$ del modelo diagonal de *Volterra* que se utiliza en este documento se usa una *PRMS* con tres niveles $u(k) = \{u_-, u_0, u_+\}$, ver la figura 3.

3.3 ENSAYO

En un primer paso se han realizado varios ensayos de la planta piloto con el objetivo de conseguir datos para la identificación de un modelo de tipo *Volterra*. Para los ensayos se ha fijado la resistencia a un valor constante del 50 % ($\rightarrow 7500$ W) y la apertura de la válvula se ha calculado mediante el método *PRMS* con tres niveles $v_8 = \{30, 60, 90\}$ %. Los ensayos se realizaron con periodos variables de la *PRMS*, pero bastante largos para observar la reacción de la planta piloto a cambios en la entrada, ver la figura 4.

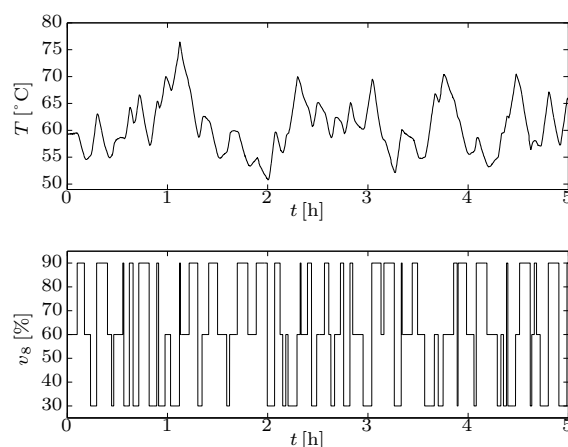


Figura 4: Resultados de ensayo de la planta piloto con la entrada calculada mediante el método de *PRMS* con tres niveles de la apertura de la válvula de recirculación $v_8 = \{30, 60, 90\}$ %.

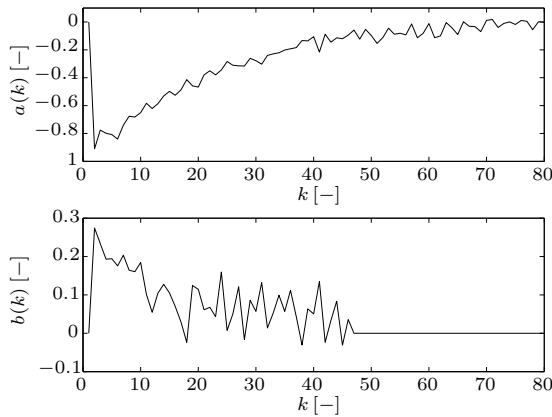


Figura 5: Parámetros identificados del modelo no lineal con $N_1 = 80$ y $N_2 = 45$.

3.4 IDENTIFICACIÓN DE PARÁMETROS

Finalmente, con los datos entrada-salida obtenidos en los ensayos, se ha realizado la identificación de los parámetros del modelo diagonal de *Volterra* de segundo orden. Para la identificación de los parámetros se ha utilizado el método de mínimos cuadrados.

Antes de identificar los parámetros, se ha normalizado la entrada de la siguiente manera:

$$\tilde{u} = \frac{u - u_0}{\Delta u} \quad (6)$$

con $u_0 = 60$ y $\Delta u = 30$ así que la entrada normalizada tiene los niveles $\tilde{u} = \{-1, 0, 1\}$.

Durante el proceso de identificación se ha observado que la elección de un tiempo de muestreo de $t_m = 30$ s representa un buen compromiso entre la bondad de la identificación y el número de parámetros. Además se ha observado que, con el tiempo de muestreo de $t_m = 30$ s, el sistema tiene un retardo de 1 periodo de muestreo.

Finalmente, se han identificado $N_1 = 80$ parámetros de primer orden y $N_2 = 45$ parámetros de segundo orden. Los parámetros identificados se pueden ver en la figura 5. El offset del modelo tiene un valor de $h_0 = 59.1$ °C.

Para evitar problemas en la posterior implementación del controlador predictivo se normaliza la salida del sistema y la referencia, de tal manera que desaparece el offset del modelo:

$$\tilde{y} = y - h_0 \quad (7)$$

$$\tilde{r} = r - h_0 \quad (8)$$

4 LEY DE CONTROL

Para predecir el futuro comportamiento del proceso se utiliza el modelo no lineal identificado

en una representación de *DMC* (Dynamic Matrix Control) [4] extendida al caso no lineal. Para el modelo de segundo orden los futuros valores de la salida vienen definidos como:

$$\tilde{y} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{c} + \mathbf{f} \quad (9)$$

con

$$\mathbf{f} = \tilde{\mathbf{u}}^T \mathbf{G}_2 \tilde{\mathbf{u}} \quad (10)$$

$$\mathbf{c} = \mathbf{H}\tilde{\mathbf{u}}_{pas} + \tilde{\mathbf{u}}_{pas}^T \mathbf{H}_2 \tilde{\mathbf{u}}_{pas} + \mathbf{d} \quad (11)$$

y las matrices \mathbf{G} (12) y \mathbf{H} (13) que representan la parte lineal del modelo identificado. Las matrices \mathbf{G}_2 y \mathbf{H}_2 consideran la parte no lineal identificada y tienen la forma que se puede ver en (14) y (15), respectivamente. El vector $\mathbf{d} = [d(k+1), d(k+1), \dots, d(k+1)]^T$ contiene la diferencia entre la salida del proceso y la salida del modelo en el instante k .

$$\mathbf{G} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ a_2 & a_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & a_1 \\ \vdots & \vdots & \ddots & a_1 + a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_P & a_{P-1} & \dots & \sum_{i=1}^{P+M+1} a_i \end{bmatrix} \quad (12)$$

$$\mathbf{H} = \begin{bmatrix} a_2 & a_3 & \dots & a_N & 0 \\ a_3 & a_4 & \dots & a_{N-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & a_1 \\ a_{P-1} & a_P & \vdots & \vdots & a_1 + a_2 \\ a_P & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & 0 & 0 \end{bmatrix} \quad (13)$$

$$\mathbf{G}_2 = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ b_2 & b_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & b_1 \\ \vdots & \vdots & \ddots & b_1 + b_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_P & b_{P-1} & \dots & \sum_{i=1}^{P+M+1} b_i \end{bmatrix} \quad (14)$$

$$\mathbf{H}_2 = \begin{bmatrix} b_2 & b_3 & \dots & \dots & b_N & 0 \\ b_3 & b_4 & \dots & b_{N-1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & b_1 \\ b_{P-1} & b_P & \vdots & \vdots & \vdots & b_1 + b_2 \\ b_P & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

Con la anterior representación del modelo no lineal se ha elegido un enfoque iterativo para calcular la acción de control. Este enfoque, presentado por Doyle *et al.* [5, 7], no considera restricciones en la acción de control. El esquema para calcular la acción de control es el siguiente:

- Paso 1: poner $i = 1$
- Paso 2: resolver las ecuaciones

$$\mathbf{a} = \left((\mathbf{r} - \mathbf{c} - \mathbf{f})^T \mathbf{G} \right)^T \quad (16)$$

$$\tilde{\mathbf{u}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{a} \quad (17)$$

- Paso 3: comprobar si la \mathbf{u} calculada cumple la condición

$$\left| \tilde{u}^{(i)}(k) - \tilde{u}^{(i)}(k-1) \right| < \delta \quad (18)$$

- Paso 4: Si la anterior condición está cumplida, poner $\tilde{u}(k) = \tilde{u}^{(i)}(k)$ y aplica $\tilde{u}(k)$. Si la anterior condición no está cumplida, recalcular \mathbf{f} usando $\tilde{u}^{(i)}(k)$, poner $i = i + 1$ y volver a paso 2.

Como no se utilizan restricciones para calcular las acciones de control se usa la saturación

$$\tilde{u}(k) = \begin{cases} \tilde{u}_{min} & \text{si } \tilde{u}(k) < \tilde{u}_{min} \\ \tilde{u}(k) & \text{en otro caso} \\ \tilde{u}_{max} & \text{si } \tilde{u}(k) > \tilde{u}_{max} \end{cases} \quad (19)$$

que limita la acción de control al intervalo $\tilde{u}_{min} \leq \tilde{u}(k) \leq \tilde{u}_{max}$.

5 RESULTADOS EXPERIMENTALES

El proceso se controla mediante el sistema de control distribuido *Simatic-IT* que incluye un Automata Programable denominado *PMC 10*. El algoritmo de control se ejecuta en *Matlab* e intercambia información con el sistema de control distribuido a través de *OPC*.

La figura 6 muestra los resultados de un ensayo de la planta piloto controlado mediante el

NMPC. Por los 80 elementos lineales del modelo no lineal identificado se ha elegido un orden de truncamiento de $N = 80$. Para el horizonte de predicción y el horizonte de control se han utilizado $P = 80$ y $M = 15$, respectivamente.

En la gráfica se pueden ver arriba la temperatura y la referencia, en el centro la acción de control y abajo las iteraciones necesarias para cumplir la condición de convergencia. La temperatura del proceso sigue bien la referencia y muestra pocas oscilaciones. Solamente en el momento de cambio de la referencia hay grandes divergencias entre temperatura y referencia, pero el controlador no lineal compensa estas divergencias en un tiempo razonable. Las acciones de control muestran considerables oscilaciones a lo largo del ensayo, incluso en instantes en los que no hay divergencias entre temperatura y referencia. En los momentos de cambio de referencia, la acción de control (la apertura de la válvula) ha sido limitada por la saturación a los valores mínimos y máximos permitidos. Con respecto a las iteraciones para cumplir la condición de divergencia se puede ver que el algoritmo necesitaba entre 3 y 11 iteraciones, que significa un cálculo rápido con los ordenadores actuales.

Además se han realizado ensayos de control predictivo lineal para obtener datos de comparación. Con el *NMPC* se ha podido reducir la suma del error cuadrático entre salida del proceso y referencia un 5.1 % frente al error del proceso controlado por un *MPC*.

6 CONCLUSIONES

En este documento se ha mostrado la aplicación de un *NMPC* basado en un modelo diagonal de *Volterra* a un proceso de laboratorio. El procedimiento de desarrollo del controlador es simple (por el uso de un modelo de *Volterra*) y rápido. El cálculo de la acción de control se ha realizado con un método iterativo que converge en pocas iteraciones y por consiguiente requiere pocos recursos. En los resultados se ha podido ver un buen comportamiento del proceso, pero también cambios permanentes en la entrada del sistema.

Para reducir estos cambios permanentes en la apertura de la válvula se puede pensar en una extensión de la ley de control, de tal manera que se incluye una ponderación de los cambios de control $\Delta \mathbf{u}$ en el algoritmo de control. Para reducir el error entre temperatura y referencia se puede incluir restricciones para la acción de control, de tal manera que se puede eliminar la saturación utilizada en la entrada del sistema. Esta medida obligaría a cambiar del algoritmo iterativo a pro-

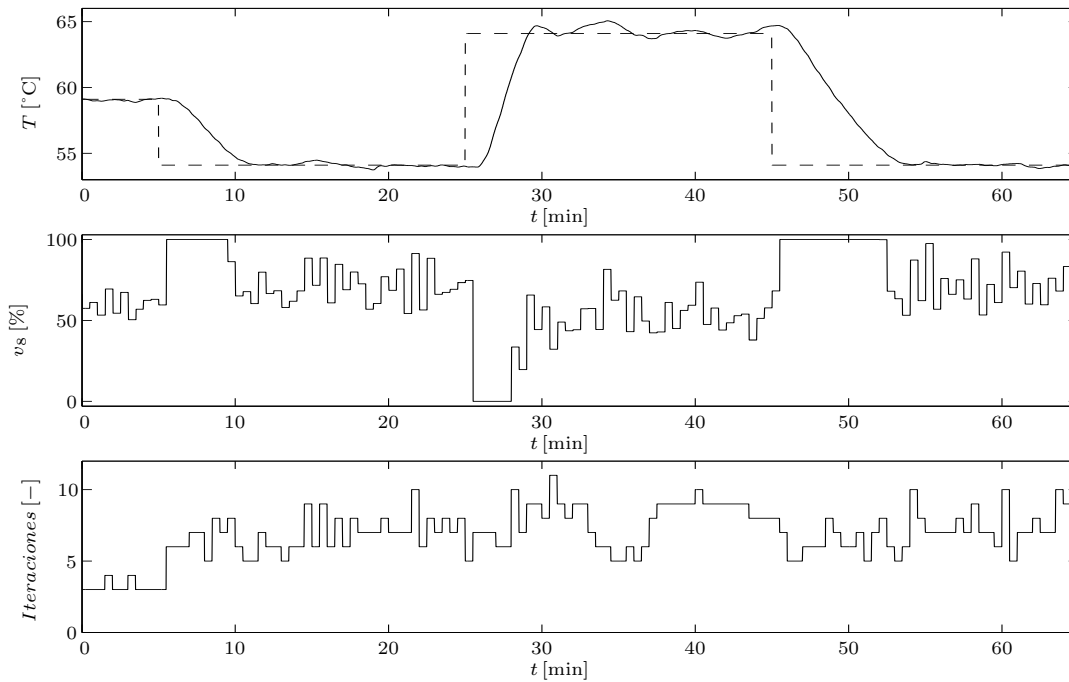


Figura 6: Resultado del ensayo de la planta piloto controlado por el MPC no lineal con cálculo iterativo de la acción de control. Arriba: la temperatura y la referencia, centro: la acción de control, abajo: las iteraciones necesarias para cumplir la condición de convergencia.

gramación cuadrática (QP).

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Educación y Ciencia a través del proyecto de investigación DPI-2004-07444-C04-01.

Referencias

- [1] Camacho, E.F., Bordons, C., (2004) "Model Predictive Control", Second Edition, Springer, London.
- [2] Cueli, J.R., (2005) "Control predictivo para procesos repetitivos", Tesis doctoral, Universidad de Sevilla.
- [3] Cueli, J.R., Bordons, C. (2005) "Iterative Nonlinear Control of a Semibatch Reactor. Stability Analysis", In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Sevilla.
- [4] Cutler, C.R., Ramaker, B.C., (1980) "Dynamic Matrix Control – A Computer Control Algorithm", In *Automatic Control Conference*, San Francisco.
- [5] Doyle, F.J., Pearson, R.K., Ogunnaike, B.A., (2002) "Identification and Control Using Volterra Models", Springer, London.
- [6] Hugo, P., Steinbach, J., Stoessel, F., (1988) "Calculation of the maximum temperature in stirred tank reactors in case of breakdown of cooling", *Chemical Engineering Science*, Vol. 43, No. 8, pp 2147-2152.
- [7] Maner, B.R., Doyle, F.J., Ogunnaike, B.A., Pearson, R.K., (1996) "Nonlinear Model Predictive Control of a Simulated Multivariable Polymerization Reactor Using Second-order Volterra Models", *Automatica*, pp 1285-1301.
- [8] Nowak, R.D., Van Veen, B.D., (1994) "Random and Pseudorandom Inputs for Volterra Filter Identification", *IEEE Transactions on Signal Processing*, Vol. 42, No. 8, pp 2124-2135.
- [9] Parker, R.S., Heemstra, D., Doyle, F.J., Pearson, R.K., Ogunnaike, B.A., (2001) "The identification of nonlinear models for process control using tailored "plant-friendly" input sequences", *Journal of Process Control*, Vol. 11, No. 2, pp 237-250.
- [10] Szeifert, F., Chovan, T., Nagy, L., (1995) "Process dynamics and temperature control of fed-batch reactors", *Computers & Chemical Engineering*, Vol. 19, No. 1, pp 447-452.

SISTEMAS DISTRIBUIDOS DE TIEMPO REAL DESDE UNA PERSPECTIVA BASADA EN SISTEMAS MULTIAGENTES CON MOVILIDAD

J.O. Coronel, J.E. Simó, A. Crespo
jacopa1@doctor.upv.es, {jsimo, alfons}@disca.upv.es

Resumen

Los sistemas de tiempo real generalmente se caracterizan porque a priori se pueden utilizar técnicas de planificación que garantizan que un conjunto de tareas será planificable durante toda su ejecución. Pero para ello también es necesario que las características de las tareas, como puede ser: el período, el plazo de ejecución, el tiempo de cómputo, el tiempo de respuesta, etc., sean conocidas anticipadamente y que además permanezcan constantes, lo cual les hace rígidos y poco adaptables a los cambios en su entorno. Este tipo de sistemas normalmente es suficiente, pero cuando en los sistemas distribuidos de tiempo real aumenta la complejidad de las aplicaciones se hace indispensable dotar al sistema de un mayor grado de adaptabilidad y flexibilidad frente a la dinámica del entorno. Desde este punto de vista, el paradigma de sistemas multiagente parece el más adecuado para afrontar la complejidad de este tipo de aplicaciones debido a su naturaleza distribuida, su autonomía y su autorregulación, y opcionalmente, es posible aumentar el nivel de flexibilidad si se considera el movimiento de agentes de tiempo real cuando las condiciones de ejecución son desfavorables en su nodo.

Palabras Clave: Sistemas Distribuidos de Tiempo Real, Planificación, Sistemas Multiagentes, Agentes Móviles de tiempo real.

1 INTRODUCCION

Los sistemas de tiempo real generalmente se caracterizan porque a priori se pueden utilizar técnicas de planificación, como las anteriormente comentadas, que garantizan que un conjunto de tareas será planificable durante toda su ejecución. Pero para ello también es necesario que las características de las tareas, como puede ser: el período, el plazo de ejecución, el tiempo de cómputo, el tiempo de respuesta, etc., sean conocidas anticipadamente y que además permanezcan

constantes, lo cual les hace rígidos y poco adaptables a los cambios de su entorno. Este tipo de sistemas normalmente es suficiente, pero cuando en los Sistemas Distribuidos de Tiempo Real (SDTR) aumenta la complejidad de sus aplicaciones se hace indispensable dotar al sistema de un mayor grado de adaptabilidad y flexibilidad frente a la dinámica del entorno.

Desde este punto de vista, el paradigma de sistemas multiagente parece el más adecuado para afrontar la complejidad de este tipo de aplicaciones debido a su naturaleza distribuida, su autonomía y su autorregulación, y adicionalmente, resulta también interesante el uso de agentes móviles de tiempo real los cuales aumentan considerablemente el grado de flexibilidad del sistema cuando las condiciones de ejecución son desfavorables.

A continuación se enunciarán algunos conceptos básicos sobre los sistemas distribuidos de tiempo real así como la justificación en la aplicación de los sistemas multiagentes a los SDTR, y todo esto basándonos ampliamente en secciones anteriores de esta tesis.

2 SISTEMAS DISTRIBUIDOS DE TIEMPO REAL

Los Sistemas Distribuidos de Tiempo Real (SDTR) están formados por un conjunto de hardware dedicado que se encuentra espacialmente distribuido y que se interconecta a través de un canal de comunicaciones compartido por el que se intercambian datos y mensajes. Normalmente este hardware son sistemas empujados que cumpliendo unos requerimientos temporales monitorizan procesos del mundo real, actúan y reaccionan a eventos. Adicionalmente, y en el caso específico de los sistemas distribuidos de control, estos sistemas empujados pueden operar en sistemas de control de lazo cerrado muestreando sensores, calculando respuestas de control y enviando esas respuestas a los actuadores correspondientes, siendo por tanto indispensable una integración determinista de la información, tanto espacial como temporal, para una correcta operación de control, tal y como es expuesto

en secciones anteriores. Este tipo de sistemas es común encontrarlos en controladores de vuelo, automóviles, robots, circuitos de vigilancia, en el control industrial, etc. en donde diferentes sensores, actuadores y dispositivos de control se encuentran dispersos. Un ejemplo gráfico de un sistema distribuido de tiempo real es mostrado en la **Figura 1: Ejemplo general de un sistema distribuido de tiempo real.**

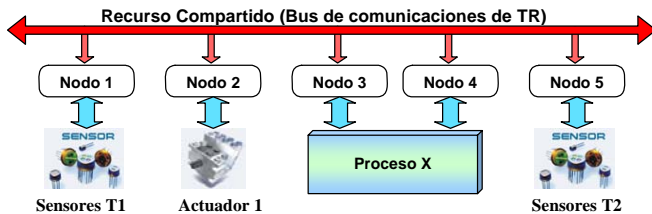


Figura 1: Ejemplo general de un sistema distribuido de tiempo real.

La naturaleza distribuida de muchas aplicaciones de tiempo real se debe a que la recopilación eficiente de información del entorno requiere una adquisición de datos desde diferentes ubicaciones físicas, un ejemplo es el mostrado en la

Figura 1.

Los elementos que componen los SDTR son inherentemente concurrentes, por lo que es necesario utilizar algún mecanismo de coordinación de acciones, y para ello, son utilizados recursos compartidos (tal como canales de comunicación). Adicionalmente y cuando es requerido algún tipo de cooperación entre componentes también se hace necesario el empleo de métodos de sincronización **[Error! No se encuentra el origen de la referencia.]**.

En los SDTR con cooperación entre nodos los requerimientos temporales se encuentran interrelacionados, de forma que una serie de eventos distribuidos deben ser ejecutados antes de un plazo de ejecución global ($D_G = D_1 + D_2 + D_3$). Es decir, los objetivos que deben ser alcanzados de forma común y en colaboración con otros elementos distribuidos tienen unos requerimientos temporales de extremo a extremo bastante críticos siendo necesario algún tipo de coordinación de elementos individuales con respecto a un D_G general que me asegure que los resultados que se generen sean temporalmente válidos.

Según las características mencionadas, en la etapa de diseño de sistemas distribuidos de tiempo real es necesario tener en cuenta las características específicas de cada nodo tales como su capacidad computacional, el tipo de sistema operativo, el tipo de algoritmo de planificación, etc., siendo además necesario analizar las características temporales de los recursos compartidos, como es el caso del bus de comunicaciones en donde es importante analizar el tipo de servicios soportados (comunicaciones *Time Triggered* o *Event Triggered*) **[Error! No se encuentra el origen de la referencia.]**.

En general, los sistemas distribuidos de tiempo real deben cumplir ciertas propiedades como son: escalabilidad, confiabilidad, integración, tolerancia a fallos, composabilidad, etc., las cuales son presentadas ampliamente en **[Error! No se encuentra el origen de la referencia.]** y **[Error! No se encuentra el origen de la referencia.]**.

2.1 SISTEMAS DISTRIBUIDOS DE TIEMPO vs SISTEMAS DE TIEMPO REAL CONVENCIONALES.

En los sistemas distribuidos de tiempo real es necesario tener en cuenta aspectos adicionales a los de los sistemas de tiempo real centralizados o convencionales tales como:

- **Planificación:** En un sistema centralizado el conjunto de tareas que se ejecutan en un procesador suele ser el único recurso compartido que se debe planificar, mientras que en los sistemas distribuidos además de lo anterior se deben planificar los mensajes que intercambian las aplicaciones, ya que estos tienen restricciones temporales. Y por consiguiente es necesario utilizar protocolos de comunicación que proporcionen algún grado de determinismo en la entrega de mensajes.
- **Asignación de tareas:** En los sistemas convencionales la ejecución de tareas se centra en la planificación de un único procesador utilizando las técnicas anteriormente estudiadas. En sistemas distribuidos se debe decidir qué tareas se ejecutan en qué nodo (procesador). Una asignación incorrecta puede llevar a que algunos procesos incumplan sus plazos (D_i).
- **Plazos (D_G) globales:** En los SDTR a diferencia de los sistemas centralizados es necesario tener en cuenta no únicamente los D individuales de un proceso o mensaje sino que hay que fijarse en el conjunto de D de los procesos y mensajes que conforman una aplicación como si fueran un todo ($D_G = D_{P1} + D_{M2} + D_{P3}$). Una aplicación distribuida puede utilizar mensajes para

coordinar la ejecución de las partes de código que se encuentren distribuidas entre distintos nodos (procesadores). Por ellos es conveniente analizar el tiempo de respuesta de la aplicación desde que comienza hasta que se termina su ejecución distribuida, teniendo en cuenta los tiempos de ejecución en cada nodo así como el tiempo de transmisión de los mensajes sobre el recurso compartido (ej. red de comunicaciones).

- **Sincronización:** como comentamos con anterioridad cuando en los SDTR se utiliza algún tipo de cooperación entre elementos es imprescindible sincronizarlos debido a que los relojes locales de cada elemento distribuido pueden tener diferentes valores, cuyas diferencias derivan de pequeñas desviaciones hardware internas. Es importante acotar la diferencia entre referencias locales y la referencia global para evitar casos en los que un determinado nodo considere un suceso como temporalmente válido y en otro nodo ese mismo suceso se interprete como fuera de plazo, dando lugar a incongruencias.
- **Tolerancia a fallos:** Una de las principales ventajas del uso de sistemas distribuidos es la abstracción de errores de tal forma que se pueda garantizar el correcto funcionamiento del sistema ante posibles fallos, abstracción que no es posible en sistemas centralizados. Las técnicas de tolerancia a fallos en SDTR consiste en la ejecución de rutinas adicionales que supervisan el funcionamiento distribuido del sistema para la detección y recuperación ante fallos, pero la ejecución de estas rutinas representan una carga extra en el sistema por lo que es posible que algún proceso incumpla sus requisitos temporales, y por consiguiente deben ser tenidas en cuenta en el diseño.

3 APLICACIÓN DE LOS SMA EN SDTR

Para el diseño de sistemas distribuidos de tiempo de real se suelen utilizar análisis estáticos de tareas y mensajes lo que puede proporcionar una idea más clara del comportamiento del sistema pero una baja tolerancia a modificaciones en tiempo de ejecución. Actualmente, las nuevas aplicaciones de SDTR empiezan a demandar nuevos requerimientos tales como flexibilidad e inteligencia, inteligencia desde el punto de vista de capacidad para adaptarse rápidamente a los cambios de su entorno, de la habilidad para comunicarse y coordinarse con otros elementos para la consecución de objetivos en común y de la capacidad de autodeterminación.

Tradicionalmente, para el desarrollo de SDTR y en general para los sistemas de tiempo real es necesario conocer con anterioridad la cantidad de tareas que se van a ejecutar al igual que sus propiedades temporales (tales como P_i , D_i , C_i) y los mensajes que se intercambiarán en el sistema distribuido para realizar un análisis y una planificación de su funcionamiento. Este procedimiento es eficiente pero bastante rígido para entornos muy dinámicos. El número de eventos externos considerados así como el número de interacciones entre nodos debe ser pequeño de tal forma que se pueda simplificar el proceso de análisis [**Error! No se encuentra el origen de la referencia.**]. Esto nos lleva a realizar ciertas asumisiones en el comportamiento del entorno que será controlado por el sistema de tiempo real, de tal forma que teóricamente el análisis de planificación nos asegurará que las tareas y los mensajes intercambiados cumplirán sus restricciones temporales durante la ejecución, pero con el aliciente de que este análisis sólo será válido mientras se cumplan las suposiciones realizadas. Si el comportamiento del entorno sufre cambios rotundos en comparación con las situaciones previstas en el diseño es posible que el sistema alcance un comportamiento indeseado, el cual puede verse reflejado en pérdidas de plazos de tareas críticas. En sistemas con entornos relativamente estáticos este riesgo puede no estar presente.

Los requerimientos inicialmente mencionados incrementan la complejidad de los sistemas distribuidos de tiempo real, complejidad que hace inviable su diseño utilizando técnicas tradicionales de planificación tal como se ha expuesto, siendo necesario recurrir a otras perspectivas que soporten las nuevas características demandadas y que además tengan en cuenta los nuevos inconvenientes que aparecerán a raíz de esto, como puede ser la inclusión o expulsión de elementos del sistema en tiempo de ejecución, la redistribución de elementos dentro del sistema distribuido, la resolución de conflictos de objetivos entre elementos, la asignación dinámica de tareas y recursos, etc. y todo esto teniendo en cuenta las restricciones temporales del sistema.

En un primer momento, el paradigma de agente y más específicamente los sistemas multiagentes (SMA) aparecen como la opción más acertada para abordar la demanda de estos nuevos requerimientos debido a que muchos de los cuales son intrínsecos en estos. Los SMA, tal como es expuesto en el capítulo 1, se caracterizan por su naturaleza distribuida, su autonomía, su autorregulación, su capacidad cooperativa, su alta adaptabilidad y opcionalmente por su movilidad, pero sin embargo, una de las características más importantes en los SDTR y de las que carecen los SMA es la gestión del tiempo como

elemento crítico en la ejecución del sistema. Por consiguiente y para una factible aplicación a los sistemas de tiempo real es necesario añadir una dimensión adicional a los SMA: el “*tiempo*”, pero el *tiempo* desde la perspectiva de los sistemas de tiempo real (T, D, C etc.). Todo esto conlleva a que los SMA deban heredar las restricciones temporales impuestas en un SDTR convencional así como el cumplimiento de algunas propiedades básicas como son: escalabilidad, integridad, disponibilidad, confiabilidad, tolerancia a fallos, etc.

La aplicación de los SMA en entornos distribuidos de tiempo real implica un cambio drástico en la filosofía de ejecución de los agentes así como en la interacción entre ellos, por lo que es posible que la mayoría de arquitecturas de agentes existentes no sean adecuadas o las que lo son soporten únicamente restricciones temporales acriticas, siendo necesario adaptarlas, o bien, siendo necesario diseñar nuevas arquitecturas, tal y como se pretende en esta tesis. A este tipo de sistemas le denominaremos “*Sistemas MultiAgente de Tiempo Real*” (SMA-TR o RT-MAS¹).

Antes de abordar las ventajas y aplicaciones que podemos conseguir con la adaptación de los SMA a entornos de tiempo real, haremos una definición formal de varios conceptos:

Definición 2.1 Agentes de tiempo real

Un agente de tiempo real es un componente software con autonomía propia que se desenvuelve en entornos con restricciones temporales y que tiene la capacidad de llevar a cabo de forma inteligente distintos objetivos mediante el intercambio de información con el entorno cumpliendo plazos estrictos de tiempo.

Podemos definir de forma general la estructura de un agente de tiempo real como:

$$A_i^{TR} = \{K_i, O_i, B_i\}$$

en donde:

$K_i \rightarrow$ constituye el conocimiento que el agente de tiempo real i tiene del entorno y cuyo conocimiento debe estar caracterizado temporalmente.

$B_i \rightarrow$ es el conjunto de comportamientos posibles del A_i^{TR} .

$O_i \rightarrow$ constituye el conjunto de objetivos que debe cumplir el agente y los cuales deben estar acotados temporalmente. Adicionalmente, para la consecución

de cada uno de los objetivos es necesario efectuar un conjunto de tareas que es posible expresar de la siguiente forma:

$$O_i = \{T_1^i, T_2^i, \dots, T_n^i\}$$

y en donde al menos una de las tareas T contiene restricciones temporales definidas de la siguiente forma:

$$T_n^i = (C_n, D_n, P_n)$$

en donde C_n es el tiempo de cómputo, D_n el plazo de entrega y P_n define el período.

Definición 2.2 Sistemas Multiagentes de tiempo real

2.1 SECCIONES Y SUBSECCIONES (10 ptos, negrita)

2.1.1 Subsección (10 ptos, negrita)

Si hubiera más subsecciones, tendrían el mismo formato que ésta subsección. Tanto los títulos de las secciones como de las subsecciones están con sangría y alineados a la izquierda.

2.2 NOTAS A PIE DE PÁGINA

No incluir cabecera ni pie de página.

2.3 CITAS EN EL TEXTO

Las citas dentro del texto deben incluir el número de referencia entre corchetes, por ejemplo [1].

2.4 FIGURAS

Todas las figuras deben estar centradas y ser claras. El número de la figura y la leyenda aparecerán siempre en la parte inferior de la figura. Cada figura debe ser numerada correlativamente y debe estar referenciada en el texto. Dejar una línea entre el texto del párrafo y la figura.

Figura 1: Ejemplo de leyenda de una figura

2.5 FÓRMULAS

Las fórmulas deben estar centradas y numeradas correlativamente, dejando una línea entre el texto y la fórmula tanto antes como después, o entre dos fórmulas consecutivas. Referenciar las ecuaciones mediante su numeración.

¹ Por el acrónimo de Real Time Multiagent System

$$\mu_A(x) = \exp\left(\frac{-(x - \mu_A(x))^2}{2\sigma_A^2}\right) \quad (1)$$

2.6 TABLAS

Todas las tablas deben estar referenciadas en el texto, como se ha hecho en la tabla 1, centradas y claras. El número y título siempre aparecerán en la parte superior de la tabla, dejando un espaciado de una línea entre el texto del párrafo y la tabla. Las tablas deben estar numeradas de manera correlativa.

Tabla 1: Ejemplo de título de una tabla.

Nombre	Descripción
A	Subconjunto borroso de X
μ_A	Función de pertenencia de A

Agradecimientos (10 ptos, negrita)

La palabra **Agradecimientos** debe ir alineada a la izquierda, no numerada y en negrita. Todos los agradecimientos deben figurar al final del trabajo.

Referencias (10 ptos, negrita)

- [1] Pedrycz, W., (1993) Fuzzy sets and fuzzy systems, Research Studies Press, England.
- [2] Zadeh, L., (1965) "Fuzzy logic", *Fuzzy Sets and Systems*, pp. 100-106.

Deben estar ordenadas por orden alfabético y justificadas con la sangría correspondiente.

SISTEMA ROBOTIZADO PARA LA DETECCIÓN DE IRREGULARIDADES EN LOSAS DE MÁRMOL Y APLICACIÓN AUTOMÁTICA DE RESINAS

Juan José Vega Gea, Manuel Cantón Garbín, Francisco Rodríguez Díaz, José Antonio Piedra Fernández

Dpto. Lenguajes y Computación. Universidad de Almería.

Carretera de Sacramento s/n. 04120. La Cañada de San Urbano, Almería (España)

juanjo.vega@gmail.com, mcanton@ual.es, frrodrig@ual.es, jpiedra@ual.es

Resumen

En la cadena de producción de losas de mármol surge el problema del sellado de los defectos localizados en las superficies de dichas losas. Hasta ahora este proceso se realiza de forma manual, pero tras los avances producidos en los últimos años, se plantea la posibilidad de sustituir dicho procesamiento manual por un sistema autónomo que se encargue de la tarea de sellado.

El objetivo es reducir los costes en tiempo, mano de obra necesaria y material empleado, centrando el sellado de los defectos en las zonas afectadas en lugar de aplicarse a toda la superficie.

El siguiente trabajo propone un método para la detección de irregularidades y defectos en las superficies de las losas de mármol mediante la utilización de técnicas de visión artificial.

Palabras Clave: visión artificial, reparación de superficies, defectos en mármol.

1 INTRODUCCIÓN

La Comarca del Mármol de Almería constituye la principal zona de extracción y transformación de mármol de Andalucía y la primera de España en cuanto a reservas, calidad del mármol y valor de la producción. Este dato es significativo por cuanto España es el segundo país productor de mármol del mundo, lo que habla por sí solo de la importancia socioeconómica del sector, tanto a nivel provincial como regional. Más de 200 canteras y otras tantas industrias de elaboración se concentran en la comarca [6].

Como ocurre con la mayor parte de sectores industriales, la industria del mármol, necesita optimizar sus procesos de producción y para ello se suele recurrir a la automatización y a la robótica.

Uno de estos procesos es el de la detección de irregularidades o defectos, y su posterior reparación, en las losas de mármol. Estos defectos son agujeros localizados sobre la superficie de las losas y su reparación consiste en taparlos aplicando resina u otro tipo de sellador sobre éstos. En la *figura 1* se puede observar una muestra de mármol defectuoso.

Este proceso, se viene realizando de forma manual, pero puede optimizarse mediante técnicas de automatización y robótica de forma que no requiera la intervención de personal alguno.

Este trabajo se centrará en las técnicas de visión artificial que permitan detectar los defectos de las superficies de las losas.

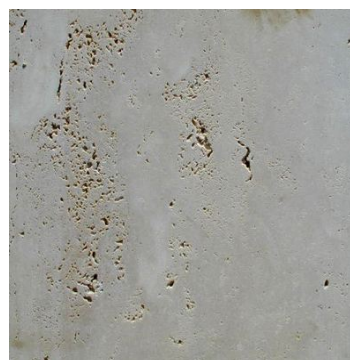


Figura 1: Ejemplo de mármol defectuoso.

2 DESCRIPCIÓN DEL SISTEMA

El sistema que se propone está dividido en dos partes: por un lado el sistema de visión y por otro el sistema robotizado de aplicación de resinas. En la *figura 2* se puede observar el sistema propuesto.

Para la construcción de un prototipo que permita comprobar el funcionamiento del sistema se ha utilizado un SCORBOT ER-IX controlado a través de un controlador tipo B que se conecta al puerto serie de un PC. Las comunicaciones se realizan mediante el estándar RS-232 utilizando un cable con conectores db9 y un esquema de conexión de módem nulo.

Se pretende detectar defectos de hasta medio milímetro, y puesto que las losas utilizadas para el desarrollo del trabajo poseen un tamaño de 30x30 cm., la resolución mínima de la cámara a utilizar debe ser de 600x600 píxels.

El sistema de visión utilizado consiste en una cámara JAI 2060 en color y una fuente de luz natural que permite iluminar la superficie uniformemente evitando reflejos. La disposición de la cámara es perpendicular a la superficie a examinar para minimizar el efecto de las distorsiones de perspectiva y el flash de la cámara permite aumentar el contraste entre la superficie y el fondo.

3 SISTEMA DE VISIÓN

Este artículo se centrará en la parte de visión: adquisición de imágenes y detección de defectos en las superficies de las losas de mármol, así como los algoritmos de procesamiento y segmentación de imágenes que se han utilizado.

Las losas de mármol utilizadas no tienen un color uniforme, lo que puede provocar errores en la identificación de los defectos y afectar al resultado final. Estas losas, generalmente grises, suelen tener partes con tonalidades blancas, amarillentas y, en algunos casos, rojas.

El principal objetivo es realzar los defectos de la superficie y suavizar al máximo las diferencias de tonalidad, permitiendo preservar el mayor número de defectos y evitando la identificación errónea de regiones de distinto color.

Las imágenes utilizadas han sido convertidas a tonos de gris utilizando el canal L del espacio HLS, ya que está demostrado que los resultados que se podrían obtener con una representación RGB son muy similares a los obtenidos mediante niveles de gris siendo el coste computacional bastante menor[5].

3.1 ALGORITMOS UTILIZADOS

3.1.1 CONVOLUCIÓN DE IMÁGENES

La convolución de imágenes se realiza mediante máscaras que tienen la forma indicada en la *tabla 1*.

Tabla 1: Máscara para el cálculo de convoluciones.

m(0,0)	m(0,1)	m(0,2)
m(1,0)	m(1,1)	m(1,2)
m(2,0)	m(2,1)	m(2,2)

Para calcular la convolución de una imagen en el caso discreto, se sitúan ambas matrices una sobre otra, se multiplica cada celda por la inferior y se

suman los productos resultantes. A continuación se desplaza la máscara a la siguiente celda y se repite el proceso hasta recorrer por completo la imagen a convolucionar [2].



Figura 2: Sistema propuesto.

Las matrices de convolución pueden tener diversos tamaños dependiendo de las necesidades, aunque el caso más común es el de matrices 3x3.

La convolución se utiliza para la aplicación de filtros como los que se mostrarán en los siguientes apartados.

3.1.2 FILTRADO DE PASO ALTO (REALCE)

El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados, bien sea por error o bien por efecto natural del método de adquisición de la imagen.

Debido a que los bordes y demás cambios bruscos de los niveles de gris están relacionados con las componentes de alta frecuencia, puede lograrse el realce de la imagen en el dominio del espacio mediante un procedimiento de filtrado de paso alto, que atenúa las componentes de baja frecuencia al mismo tiempo que intensifica los detalles [3].

Para realizar el filtrado de una imagen se procede a calcular la convolución de ésta con una matriz

cuadrada del orden deseado (también denominada máscara).

La máscara utilizada para implementar un filtro espacial de paso alto debe tener coeficientes positivos cerca de su centro y negativos en la periferia. Para una máscara 3x3 esta condición se cumple escogiendo un valor positivo en el centro y negativos en el resto [3].

En este caso, la matriz de convolución para el filtrado de paso alto es la que se muestra en la *tabla 2*. Tras realizar varias pruebas se escogió aquel filtro que proporcionaba un realce suficiente para las siguientes fases del proceso.

Tabla 2: Matriz de convolución para el filtrado de paso alto.

-1	-1	-1
-1	10	-1
-1	-1	-1

En la *figura 3* se puede observar el resultado del filtrado de paso alto y, como se puede apreciar, los defectos de la superficie se muestran mucho más nítidos que en la imagen original.

3.1.3 FILTRADO DE LA MEDIANA

Se trata de un filtro no lineal en el dominio espacial y se utiliza para la eliminación de ruido, es decir, para suavizar píxeles que difieren mucho de sus vecinos.

El nivel de gris de cada píxel es sustituido por la mediana de los niveles de gris de un entorno de tamaño dado de este píxel.

La mediana de un conjunto es el valor tal que la mitad de los valores de dicho conjunto son menores que él y la otra mitad son mayores.

El filtrado de la mediana es muy efectivo para eliminar ruido de una imagen ya que permite deshacerse de píxeles aislados que por su reducido tamaño son irrelevantes en el proceso.

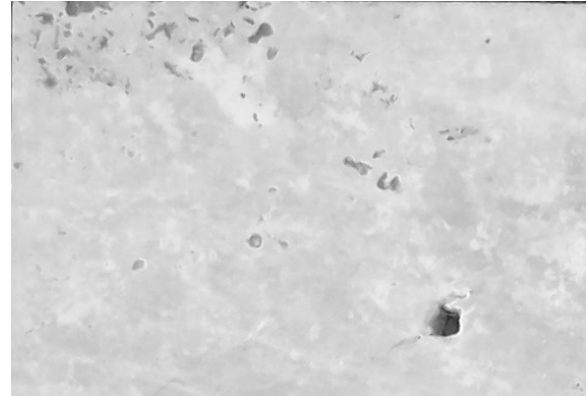
3.1.4 EROSIÓN Y DILATACIÓN

Se denomina transformaciones morfológicas a aquellas que modifican la estructura o forma de los objetos que están presentes en una imagen. Estas transformaciones son una herramienta muy útil para la extracción de características además de permitir la eliminación de ruido.

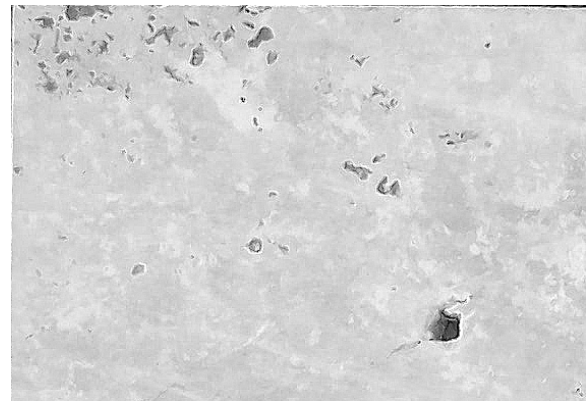
Las dos operaciones morfológicas principales son la erosión y la dilatación [2].

• **Erosión.** Para los conjuntos A y B de Z^2 , la erosión de A por B, representada por $A \ominus B$, se define como:

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (1)$$



b) Imagen original.



b) Imagen filtrada.

Figura 3: Ejemplo de filtrado de paso alto.

Es decir, la erosión de A por B es el conjunto de todos los puntos x tales que B, trasladado por x, está contenido en A.

El conjunto B recibe el nombre de *elemento estructurante*.

• **Dilatación.** Siendo A y B conjuntos de Z^2 y \emptyset representando al conjunto vacío, la dilatación de A por B, representada por $A \oplus B$, se define como:

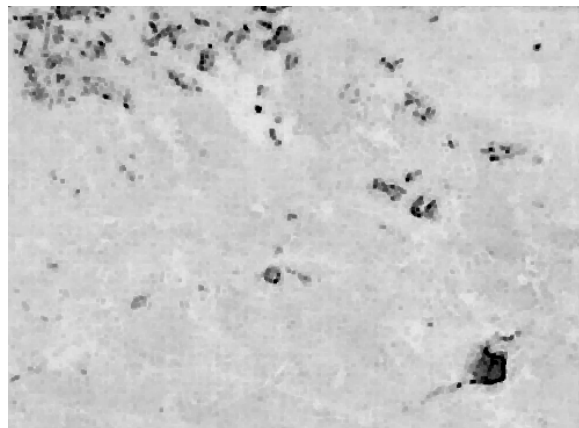
$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} \quad (2)$$

La dilatación de A por B es entonces el conjunto de todos los desplazamientos x tales que \hat{B} y A se solapan en al menos un elemento distinto de cero.

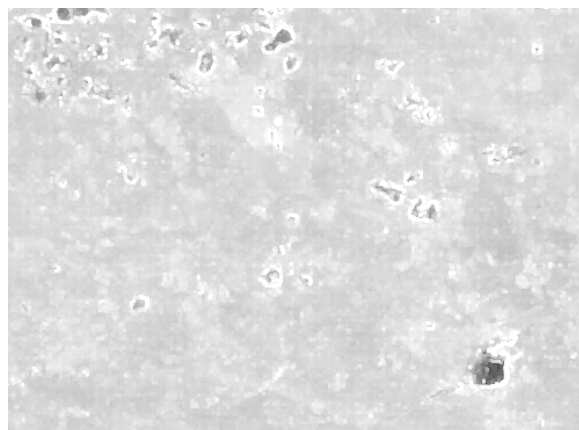
La *figura 4* muestra un ejemplo de erosión y dilatación sobre la imagen de la *figura 3.a*. Como puede observarse, las regiones claras se agrandan en

el caso de aplicar dilatación y se encogen en caso de aplicar erosión.

Estas dos operaciones son la base para la apertura y el cierre, las cuales se mostrarán en el siguiente apartado.



a) Erosión.



b) Dilatación.

Figura 4: Ejemplo de erosión y dilatación.

3.1.5 APERTURA Y CIERRE

Las dos transformaciones anteriores no son conmutativas, es decir, después de una erosión no se puede recuperar la imagen original con una dilatación ya que se habrán perdido pequeños detalles. Sin embargo, esto posee una gran utilidad puesto que la información que se mantiene corresponderá a las partes más importantes de los objetos y además se habrán eliminado pequeños elementos, reduciéndose el ruido [2].

• **Apertura.** La apertura se define como una combinación de erosiones y dilataciones en las que la primera operación es una erosión y la última una dilatación, utilizando siempre el mismo elemento estructural:

$$A \circ B = (A \ominus B) \oplus B \quad (3)$$

Se denomina apertura ya que al comenzar con una erosión se tiende a romper las piezas en sus partes constitutivas. [2]

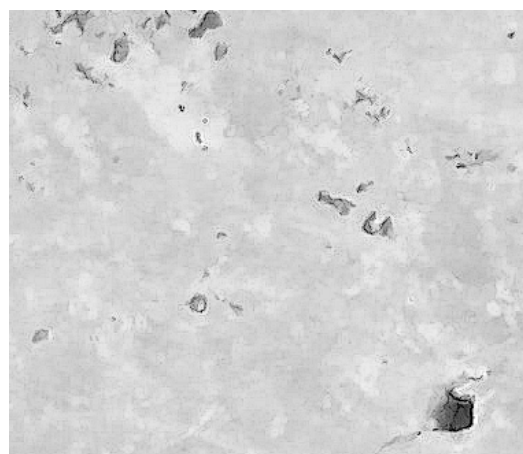
La apertura suaviza los contornos de los objetos y elimina protuberancias finas.

• **Cierre.** El cierre es la operación dual de la apertura y su definición es la siguiente [2]:

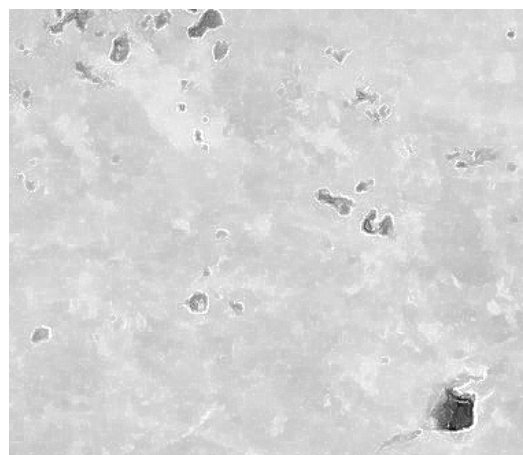
$$A \bullet B = (A \oplus B) \ominus B \quad (4)$$

Al igual que la apertura, el cierre también suaviza los contornos, pero generalmente fusiona las hendiduras finas y largas presentes en los objetos, elimina agujeros pequeños y rellena brechas en el contorno.

La figura 5 muestra un ejemplo de apertura y cierre sobre la imagen original mostrada en la figura 3.a. Se puede observar como los cambios de color tienen a uniformarse tras la aplicación de la operación de apertura y como las regiones pequeñas tienden a desaparecer tras la aplicación del cierre.



a) Apertura.



b) Cierre.

Figura 5: Ejemplo de apertura y cierre.

3.1.6 UMBRALIZACIÓN

Uno de los métodos más utilizados en la segmentación de imágenes es la *umbralización*.

Si se dispone de una imagen con objetos luminosos sobre un fondo oscuro, el histograma de la imagen mostrará dos grupos principales como se muestra en la *figura 6*.

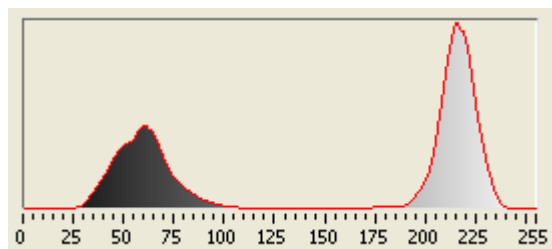


Figura 6: Histograma.

Con un histograma como el de la figura anterior, se puede seleccionar un umbral T que separe ambos grupos, de forma que cualquier punto (x,y) para el que $f(x,y) > T$ será un punto del objeto mientras que en caso contrario se tratará de un punto del fondo. La elección del umbral T adecuado es un aspecto decisivo para obtener los resultados deseados [3].

3.2 IDENTIFICACIÓN DE REGIONES

Una vez que se ha optimizado la imagen para realzar y detectar después los defectos, se hace necesaria la interpretación de ésta identificando las regiones que la componen.

El objetivo de la identificación de regiones es detectar el área mínima que encierra un conjunto de puntos próximos entre sí. De esta forma el robot no tendrá que sellar cada punto de forma independiente si no que se aplicará el sellador a grupos de puntos.

En la *tabla 3* se muestran las coordenadas del rectángulo mínimo que contiene a cada una de las regiones. Se indican dos puntos: el superior derecho y el inferior izquierdo. La *figura 7* muestra el resultado de la representación de las regiones para una sección de una de las imágenes de prueba.

Tabla 3: Tabla de características de las regiones

Objeto nº	X Superior-Izquierda	Y Superior-Izquierda	X Inferior-Derecha	Y Inferior-Derecha
1	532,000	52,000	535,000	54,000
2	523,000	54,000	529,000	57,000
3	990,000	54,000	993,000	56,000
4	375,000	57,000	377,000	59,000
5	422,000	57,000	428,000	62,000
...				

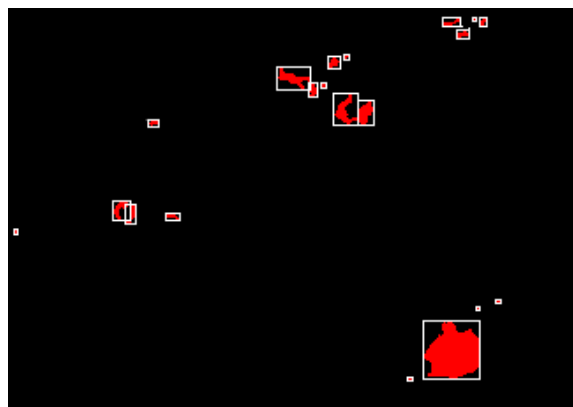


Figura 7: Identificación de regiones.

3.3 IMPLEMENTACIÓN

La implementación de la metodología propuesta se ha llevado a cabo mediante LabVIEW 7.1 [4] y se muestra en la *figura 8*.

Básicamente los pasos que se realizan son los siguientes:

1. **Conversión de la imagen a escala de grises.** Para ello se extrae el canal L del espacio HLS.
2. **Realce de detalles.** Se aplica el filtro mostrado en la *tabla 2* para realzar los detalles de la imagen. Esto provoca que pequeños defectos se muestren con mayor claridad.
3. **Transformación mediante apertura.** Como se ha mostrado en apartados anteriores, la apertura permite que algunas regiones muy cercanas se fusionen, lo que posteriormente permitirá tratar nubes de puntos muy cercanos como una única región.
4. **Umbralización.** Mediante la técnica de umbralizado se pueden aislar los defectos de la superficie de la losa. El umbral elegido determina la precisión que tendrá la detección, así como el grado de error que se cometa ya que se pueden interpretar como defectos algunas variaciones de color.
5. **Filtrado de la Mediana.** Mediante la aplicación de un filtro de la mediana se consigue un suavizado general de la imagen que permite la eliminación de pequeños puntos. De esta forma como resultado final se obtendrán las regiones principales.

4 RESULTADOS PRELIMINARES

A continuación se mostraran algunos de los resultados obtenidos con las técnicas propuestas.

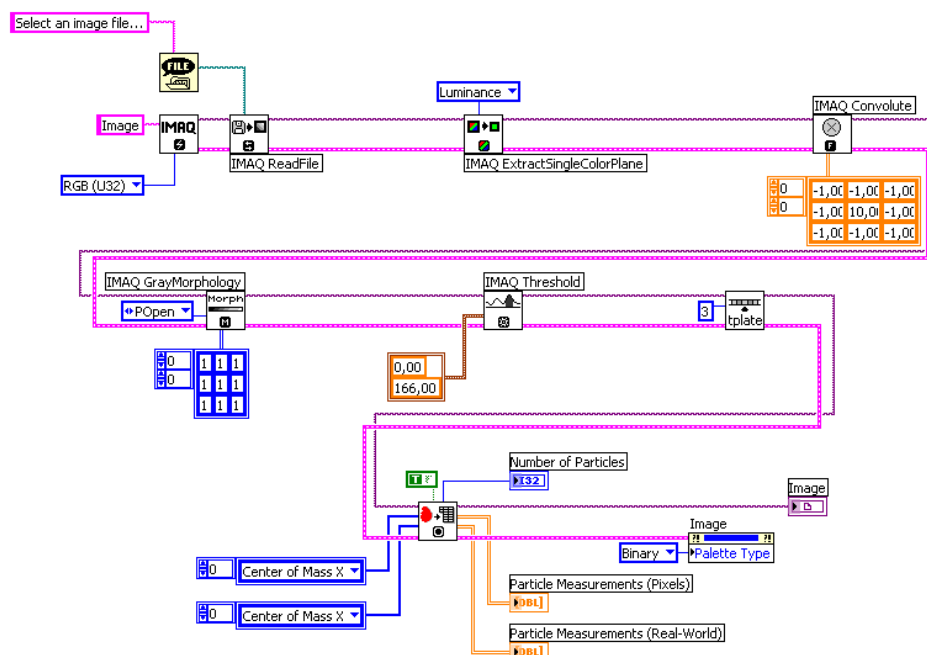


Figura 8: Implementación del sistema de visión.

4.1 IDENTIFICACIÓN DE DEFECTOS

Como se puede observar en la *figura 9*, el sistema es capaz de detectar distintas cantidades y tamaños de defectos independientemente de la densidad de estos.

Los cambios de orientación en las losas o la ubicación de los defectos no influyen en el resultado final de la interpretación. En la *figura 10* se puede observar el resultado tras girar la imagen de la *figura 9* a 90° en el sentido de las agujas del reloj.

4.2 LOCALIZACIÓN DE DEFECTOS

El proceso de reconocimiento completo consiste en una serie de pasos que se suceden secuencialmente.

Como ya hemos visto, lo primero que se hace es mejorar la imagen para favorecer los aspectos que nos interesan, los defectos de la superficie de las losas, y a continuación se segmenta la imagen para aislar las regiones de interés.

Después se procede a identificar y extraer las características de las regiones que hayan aparecido tras el paso anterior.

De la identificación de regiones se extrae una tabla con las características de éstas, lo que permitirá indicar al robot manipulador el trabajo a realizar.

La *figura 11* muestra el proceso de reconocimiento completo.

5 CONCLUSIONES Y FUTUROS TRABAJOS

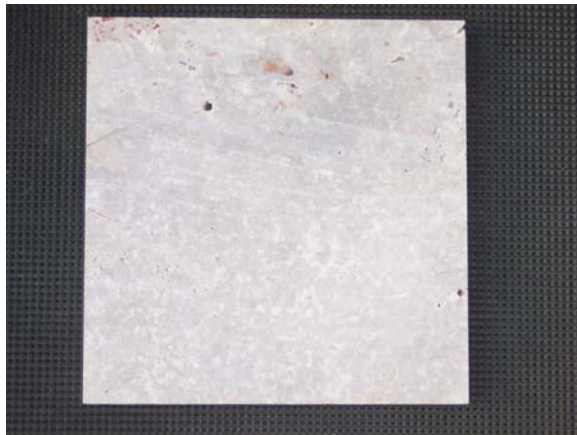
En el presente trabajo se ha expuesto un estudio sobre la detección de irregularidades y defectos en las superficies de las losas de mármol utilizando diversos algoritmos para el tratamiento de imágenes.

La principal ventaja del sistema propuesto está en el ahorro de material empleado en sellar los defectos de las losas. Si en el proceso manual se procedía a cubrir la mayor parte de la superficie, el sistema propuesto aplicará el sellador de forma localizada, es decir, en aquellas zonas que hayan sido detectadas tras el proceso de identificación.

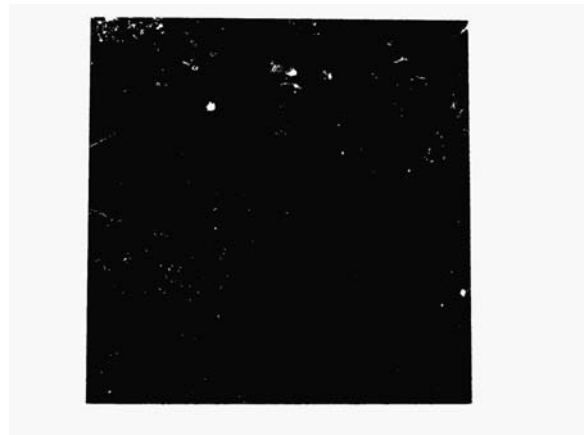
Con las pruebas realizadas se ha comprobado que por lo general el área de superficie afectada por defectos se encuentra en un rango de unos 600.000 a 1.200.000 píxeles, por lo que si las imágenes tienen aproximadamente unos 3.000.000 de píxeles, el sistema propuesto provoca que solamente se cubra entre un 20% y un 40% de la superficie total, o lo que es lo mismo, supone un ahorro de entre el 60% y el 80% de material de sellado.

Queda pendiente estudiar la propuesta de Birginia y Rivasb [1] sobre la utilización de escáneres láser para resaltar la degradación de las superficies de piedra en condiciones climatológicas adversas.

En el futuro se pretende desarrollar un sistema completo que disponga de un sistema de adquisición de imágenes propio y que integre las técnicas de visión con el SCORBOT ER-IX para el sellado automático de los defectos de las losas.



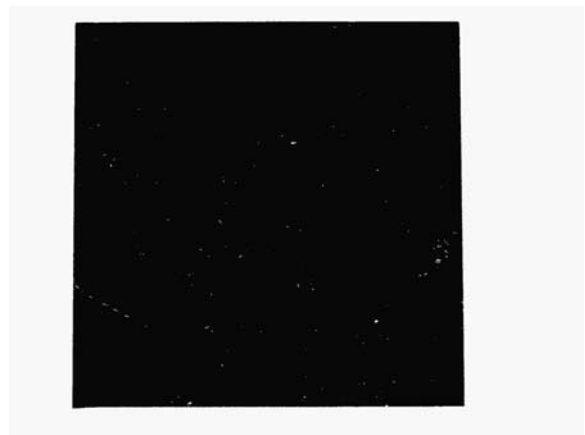
a) Imagen original 1.



b) Resultado 1.

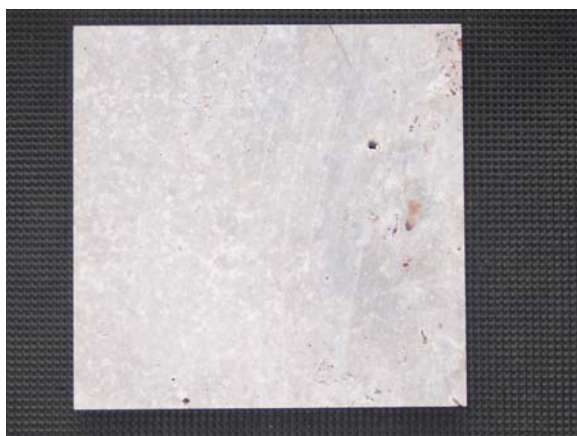


c) Imagen original 2.

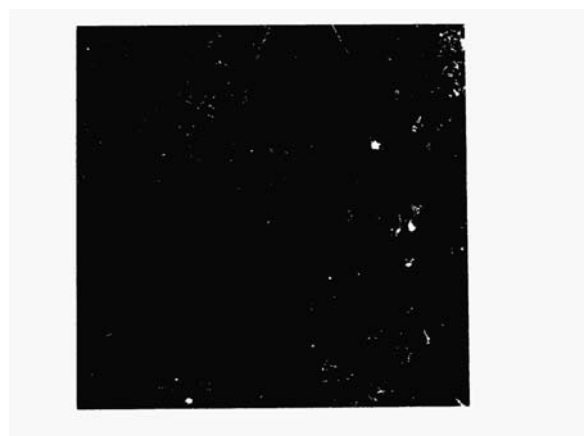


d) Resultado 2.

Figura 9: Ejemplo de distintas densidades de defectos.



a) Imagen original 1 girada 90°.



b) Resultado.

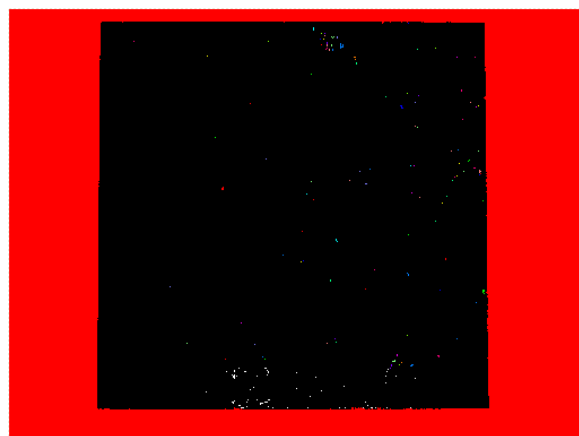
Figura 10: Ejemplo de cambio de orientación de defectos.



a) Imagen original.



b) Preprocesamiento.



c) Reconocimiento de regiones.

Objeto nº	Primer Pixel X	Último Pixel Y	Área
1	53.200.000	5.200.000	500.000
2	52.300.000	5.400.000	1.400.000
3	99.000.000	5.400.000	500.000
4	37.500.000	5.700.000	300.000
...			

d) Características de las regiones.

Figura 11: Proceso de reconocimiento.

Referencias

- [1] Birginia, J. M.; Rivasb, T.; Use of a laser camera scanner to highlight the surface degradation of stone samples subjected to artificial weathering. www.sciencedirect.com
- [2] de la Escalera Hueso, A.; (2001) Visión por Computador, Fundamentos y métodos. Prentice-Hall.
- [3] González, Rafael C. (1996) Tratamiento digital de imágenes. Addison-Wesley / Díaz de Santos.
- [4] Lázaro, Antoni Manuel; Del Río Fernández, Joaquín. LabVIEW 7.1: programación gráfica para el control de instrumentación. Thomson-Paraninfo, 2005.
- [5] Ortiz, F.; Corrales, Juan A.; Torres, F.; (2005) Clasificación de la piedra de mármol mediante morfología matemática en color. XXVI Jornadas de Automática.
- [6] Villalobos Megía, Miguel; Cañete Pérez, José A.; Salas Martín, Ricardo. El Sector Del Mármol en la Provincia de Almería. Perspectivas Económicas Y Problemática Medioambiental.
<http://www.gem.es/materiales/document/documen/g04/d04101/d04101.htm>

Cooperación y percepción cooperativa en el sistema multi-UAV COMETS

Luis Merino

Universidad Pablo de Olavide, Crta. Utrera, km. 1, 41013 Sevilla, lmercab@upo.es

Aníbal Ollero, Fernando Caballero

Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092 Sevilla {aollero,caba}@cartuja.us.es

Grupo de Robótica, Visión y Control <http://grvc.us.es>

Resumen

El artículo presenta resultados obtenidos en el proyecto europeo COMETS en el cual se ha desarrollado y demostrado un sistema para la coordinación y control de múltiples vehículos aéreos heterogéneos. Este artículo introduce el sistema COMETS y presenta métodos de percepción cooperativa multi-UAV con aplicación a la detección y localización de alarmas. Se incluyen resultados de experimentos de detección, confirmación y localización precisa de incendios forestales.

Palabras clave: robótica aérea, coordinación de robots, percepción cooperativa, detección incendios.

1. INTRODUCCIÓN

Los equipos de múltiples robots han sido objeto de una atención creciente durante la última década. En la mayor parte de los trabajos se consideran equipos de robots terrestres, y fundamentalmente en interiores.

En el caso de vehículos aéreos no tripulados (UAVs en sus siglas en inglés), los esfuerzos se han concentrado en la coordinación de vehículos del mismo tipo para actividades como el vuelo en formación [4]. También se ha investigado la coordinación de vehículos aéreos y terrestres [12] [11].

El objetivo del proyecto COMETS era el desarrollo de una arquitectura para el control y la coordinación de múltiples UAVs heterogéneos. La heterogeneidad abarca desde el nivel de autonomía hasta las plataformas consideradas. En este artículo se muestran algunos aspectos de la arquitectura desarrollada, para a continuación centrarse en la cooperación para la percepción.

En cuanto a sus aplicaciones, los UAVs han sido fundamentalmente usados en el campo militar. Más recientemente se han presentado los primeros resultados en aplicaciones tales como la detección de minas [1].

El sistema desarrollado en el proyecto COMETS puede ser empleado en numerosas aplicaciones de



Figura 1: Imagen de un incendio desde un UAV.

vigilancia y monitorización. En el proyecto se ha incluido la demostración del sistema en la detección y monitorización de incendios forestales (figura 1), problema particularmente relevante en numerosas regiones de nuestro planeta, incluyendo el sur de Europa. En el proyecto COMETS se ha realizado la primera demostración de un sistema multi-UAV en experimentos de fuegos forestales.

El artículo se organiza del siguiente modo. La sección 2 analiza algunos de los aspectos fundamentales en la coordinación de UAVs heterogéneos. A continuación, la sección 3 presenta las técnicas empleadas para la percepción cooperativa entre varios UAVs. La sección 4 presenta algunos de los experimentos de fuego en que el sistema ha sido demostrado, así como resultados concretos obtenidos en dichos experimentos. Finalmente, se aportan conclusiones.

2. COORDINACIÓN DE MÚLTIPLES UAVS. EL SISTEMA COMETS

Una arquitectura para la coordinación de múltiples robots debe considerar los siguientes aspectos:

- **Supervisión y ejecución de tareas.** La ejecución de tareas es un proceso pasivo, que se encarga del manejo de la ejecución de las distintas tareas, mientras que la supervisión

es un proceso activo que controla todas las actividades deliberativas del robot.

- **Coordinación de tareas.** La coordinación asegura la coherencia en las actividades que se desarrollan en un grupo de robots. Se deben definir mecanismos para la resolución de conflictos, especialmente para el caso de planificación de trayectorias y la ejecución de tareas cooperativas (por ejemplo, la percepción simultánea de un mismo objetivo por parte de varios robots).
- **Planificación y refinamiento de misiones.** Estas actividades consideran todo lo que es la construcción de planes coherentes, considerando para ello modelos de las tareas y modelos del entorno.
- **Asignación de tareas.** Es necesario abordar cómo se distribuyen las tareas entre los robots. Requiere establecer un protocolo de asignación de tareas, así como definir métricas que permitan cuantificar la relevancia de asignar una tarea a uno u otro robot.

La arquitectura desarrollada en COMETS aborda todos estos aspectos, a los que se añade una particularidad adicional. Los UAVs considerados en COMETS son heterogéneos también en el grado de autonomía de cada uno de ellos. Por tanto, la arquitectura COMETS es capaz de coordinar al mismo tiempo sistemas teleoperados, sistemas con autonomía operacional (las tareas se planifican y supervisan en un centro de control) y sistemas con autonomía de decisión (capaces de realizar misiones complejas por ellos mismos). Además, dicha arquitectura da soporte a que el centro de control tome el gobierno de los UAVs con autonomía de decisión, para hacer frente a posibles situaciones de emergencia por ejemplo.

Por tanto, la arquitectura de COMETS permite integrar tanto planificación puramente centralizada como sistemas distribuidos de decisión, en función del contexto y la misión a desarrollar. Estos aspectos hacen de ella un sistema multi-robot versátil y flexible. Más detalles pueden encontrarse en [3], [2].

3. PERCEPCIÓN COOPERATIVA ENTRE MÚLTIPLES UAVS

Uno de los objetivos fundamentales de COMETS era el desarrollo de técnicas para la percepción cooperativa entre los UAVs. La percepción cooperativa puede ser definida como la colaboración de

los robots de un equipo para la estimación del entorno, mediante la fusión de datos obtenidos por diferentes robots e incluso mediante la ejecución de acciones de forma coordinada.

Por tanto, el sistema de percepción cooperativa debe tratar con un aspecto fundamental: la fusión de datos. Ésta es necesaria para aumentar la fiabilidad en la estimación al emplear distintas fuentes de información. Además, en una flota de UAVs heterogéneos, los vehículos poseerán sensores de distinto tipo. Por tanto, los algoritmos de fusión deben ser capaces de admitir información de distinta naturaleza.

Esta sección presenta el esquema general de este sistema para el caso de la detección, confirmación y localización de eventos de interés por parte de una flota de UAVs. Para clarificar la exposición, se considerará su aplicación concreta a la detección de incendios.

3.1. DETECCIÓN Y LOCALIZACIÓN DE ALARMAS EMPLEANDO GRIDS

Una de las misiones fundamentales de la flota de UAVs es la detección y localización de eventos de interés. El sistema de percepción cooperativa debe hacer uso de los distintos sensores en la flota para determinar las posiciones de las posibles alarmas, reduciendo en lo posible el número de falsas alarmas.

Esta sección describe el algoritmo empleado para la detección y localización cooperativa. La única premisa necesaria es un mapa de alturas de la zona a explorar (que podría ser generado por otro de los UAVs, como Karma [5]). La superficie 3D se representa mediante una matriz 2D o grid. Cada celda i del grid tiene una posición 3D asociada \mathbf{x}_i y una probabilidad $p(h_i)$ de contener un evento de interés (por ejemplo, fuego en la aplicación de detección de incendios). Es decir, cada celda tiene asociada una variable aleatoria binaria h_i , que sigue una distribución de Bernoulli. Representaremos mediante h_i el hecho de que haya un objeto en la celda i y mediante \bar{h}_i el contrario, de modo que $p(h_i) = 1 - p(\bar{h}_i)$.

Las probabilidades asociadas a las celdas del grid son actualizadas a medida que se obtienen nuevos datos usando un ciclo de predicción-actualización.

Sea $\mathbf{S}^{k+1} = \{\mathbf{S}_0, \dots, \mathbf{S}_{k+1}\}$ el conjunto de todos los datos obtenidos hasta el instante $k+1$. El objetivo es estimar, en el instante $k+1$, la función de distribución *a posteriori* $p(h_{i,k+1} | \mathbf{S}^{k+1})$ para todas las celdas del grid; es decir, la probabilidad de que exista un evento en cada celda i en el instante $k+1$, condicionada a los valores de los datos.

Estos datos consistirán en las lecturas de los distintos sensores de los UAVs (como las cámaras y otros sensores).

La función de distribución se puede escribir como:

$$p(h_{i,k+1}|\mathbf{S}^{k+1}) = p(h_{i,k+1}|\mathbf{S}_{k+1}, \mathbf{S}^k) \quad (1)$$

Usando la regla de Bayes:

$$p(h_{i,k+1}|\mathbf{S}_{k+1}, \mathbf{S}^k) = \frac{p(\mathbf{S}_{k+1}|h_{i,k+1}, \mathbf{S}^k)p(h_{i,k+1}|\mathbf{S}^k)}{\eta} \quad (2)$$

donde

$$\eta = p(\mathbf{S}_{k+1}|h_{i,k+1}, \mathbf{S}^k)p(h_{i,k+1}|\mathbf{S}^k) + p(\mathbf{S}_{k+1}|\bar{h}_{i,k+1}, \mathbf{S}^k)p(\bar{h}_{i,k+1}|\mathbf{S}^k) \quad (3)$$

y $p(\bar{h}_{i,k+1}|\mathbf{S}^k) = 1 - p(h_{i,k+1}|\mathbf{S}^k)$. Se asume que las medidas tomadas en distintos instantes son condicionalmente independientes, de modo que

$$p(\mathbf{S}_{k+1}|h_{i,k+1}, \mathbf{S}^k) = p(\mathbf{S}_{k+1}|h_{i,k+1}) \quad (4)$$

y del mismo modo para $p(\mathbf{S}_{k+1}|\bar{h}_{i,k+1}, \mathbf{S}^k)$. El término $p(h_{i,k+1}|\mathbf{S}^k)$ se calculará a partir de la información de la que se dispone en el instante k , $p(h_{i,k}|\mathbf{S}^k)$ usando (5):

$$p(h_{i,k+1}|\mathbf{S}^k) = \sum_j p(h_{i,k+1}|h_{j,k})p(h_{j,k}|\mathbf{S}^k) \quad (5)$$

donde $p(h_{i,k+1}|h_{j,k})$ es la probabilidad de transición que relaciona la probabilidad de que exista un evento en la celda i si hay un evento en la celda j el instante anterior. Por tanto, el modelo de transición considera que $h_{i,k+1}$ depende de todo el conjunto de celdas en el instante anterior, y no sólo en el estado de ella misma $h_{i,k}$. De este modo, la probabilidad de transición $p(h_{i,k+1}|h_{j,k})$ permite considerar la propagación del evento de unas celdas a otras. El modelo concreto depende de la aplicación. Para la detección de incendios, y si nada más se conoce, (por ejemplo, la dirección del viento, que podría ser incluida en el modelo), este modelo de transición considera que el fuego se propaga con igual probabilidad en todas direcciones, con lo que consiste en un suavizado de las probabilidades de las celdas. Otros aspectos podrían ser tenidos en cuenta en este modelo de transición, como por ejemplo la pendiente del terreno en cada celda.

Por tanto, mientras los UAVs no proveen nuevos datos, el estado de las alarmas hasta el instante $k+1$ viene dado por la ecuación (5). Cada vez que se reciben nuevos datos desde un UAV, el estado es actualizado usando (2).

El término $p(\mathbf{S}_{k+1}|h_{i,k+1})$ es la función o modelo del sensor, la función de verosimilitud. Determina la probabilidad de obtener los datos \mathbf{S}_{k+1} si se tiene un objeto de interés en la celda i en el instante $k+1$. Los datos \mathbf{S}_{k+1} consisten en toda la información recogida por los robots del equipo en el instante $k+1$. Las medidas de los distintos sensores se consideran condicionalmente independientes dado el valor de las celdas, y, por tanto:

$$p(\mathbf{S}_{k+1}|h_{i,k+1}) = \prod_j p(\mathbf{S}_{j,k+1}|h_{i,k+1}) \quad (6)$$

con j un índice sobre todos los sensores que obtienen datos en el instante $k+1$.

Los diferentes términos de (6) considerarán las posiciones de los distintos sensores respecto del mapa, así como las relaciones geométricas entre los puntos en el espacio y las lecturas de los mismos. El modelo de esto último se obtendrá mediante procesos de calibración, mientras que la posición de los sensores será calculada por los propios robots. Es necesario tener en cuenta las incertidumbres asociadas para actualizar correctamente los valores de las celdas del grid.

En el instante $k+1$, y para cada sensor j , el hardware a bordo de los UAVs permite obtener una estimación de la posición $\mathbf{t}_{j,k+1}$ y orientación $\mathbf{R}_{j,k+1}$ del sensor en un sistema de coordenadas global a toda la flota. Denominaremos mediante $\mathbf{q}_{j,k+1}$ a estos datos de posicionamiento. Cada UAV también obtiene una estimación de los errores en dichas cantidades (la matriz de covarianza de los errores). Puesto que las medidas obtenidas por los sensores dependen de la posición de los mismos, estos errores deben ser tenidos en cuenta al calcular la verosimilitud de los datos. Por tanto, cada factor en (6) consiste en (a partir de este momento, el índice temporal no se muestra salvo que sea necesario):

$$p(\mathbf{S}_j|h_i) = \int p(\mathbf{S}_j|h_i, \mathbf{q}_j)p(\mathbf{q}_j)d\mathbf{q}_j \quad (7)$$

El cálculo de $p(\mathbf{S}_j|\bar{h}_i)$ se realiza mediante expresiones equivalentes a (6) y (7).

3.2. MODELOS DE LOS SENSORES

Los distintos UAVs de la flota pueden llevar a bordo distintos sensores. Para poder aplicar las técnicas anteriores, la función de verosimilitud de los distintos sensores debe ser caracterizada. Este apartado va a describir la función para los sensores empleados en la misión de detección de fuegos.

Uno de los vehículos, el helicóptero autónomo Marvin de la Universidad Técnica de Berlín posee un sensor detector de fuegos, consistente en



Figura 2: Imagen infrarroja de una escena con un fuego capturada por Heliv.

un foto-detector ajustado para responder ante las radiaciones ultravioleta propias del fuego [10].

El helicóptero Heliv lleva a bordo una cámara visual y una cámara infrarroja [8]. La figura 2 muestra una de las imágenes infrarrojas tomadas desde dicho vehículo. También, tanto Heliv como Marvin poseen cámaras de vídeo en color .

Los sensores considerados tienen muy diferentes características en lo que respecta a la localización, es decir, la cantidad de información que proporcionan acerca de la posición del fuego. En el caso del sensor de fuegos, esta información es pobre, pues sólo indica la presencia o ausencia de fuego en el campo de visión del sensor. En el caso de las cámaras esta información es algo mejor, pues las cámaras indican la dirección en la que se encuentra el fuego.

Cuadro 1: Caracterización datos preprocesados

	Sensor de fuego	IR	Visual
P_D	95 %	100 %	90 %
P_F	5 %	10 %	3 %

Los datos procedentes tanto de las cámaras como del sensor de fuego son preprocesados en una primera fase. Para el caso del sensor de fuego presente en Marvin, el preprocesamiento consiste en una umbralización, de modo que el resultado final es un valor binario indicando si hay fuego presente dentro de su campo de visión. Para las imágenes (tanto infrarrojas como visuales), el preprocesamiento consiste en una segmentación. Usando estos algoritmos, ambos tipos de imágenes son transformados en imágenes binarias en las que los píxeles se clasifican como fuego o no fuego. Dichos algoritmos no se describirán en el presente documento (ver [7] y [9] para más detalles).

Tanto los algoritmos de segmentación como la umbralización para el sensor de fuego han sido caracterizados en función de sus probabilidades P_D de detección y P_F de falso positivo. La tabla 1 muestra los valores típicos.

Las ecuaciones de medida tendrán en cuenta las

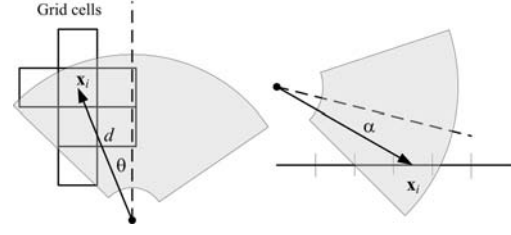


Figura 3: Esquema del modelo de medida del sensor de fuego.

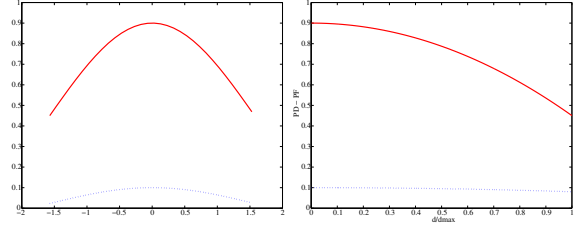


Figura 4: Gráficas correspondientes a (8) y (9). Izquierda, componente angular. Derecha, componente debida a la distancia.

características de este preprocesamiento. A continuación se muestra en detalle los modelos de ambos sensores.

3.2.1. Sensor de fuego

Como se ha dicho, tras un pre-procesamiento, el sensor de fuego proporciona un valor binario indicando la presencia o ausencia de fuego en su campo de visión. Este campo de visión viene determinado por dos ángulos de apertura (horizontal y vertical) y una distancia máxima (Figura 3).

El modelo $p(\mathbf{S}_j|h_i, \mathbf{q}_j)$ está determinado por las probabilidades $P_{D,j}$ y $P_{F,j}$ de la Tabla 1. Estas probabilidades nominales se modifican en función de la posición relativa de la celda i , \mathbf{x}_i , respecto de la posición y orientación del sensor (\mathbf{q}_j), y así:

$$P_{D,j}(\mathbf{x}_i, \mathbf{q}_j) = P_{D,j} - w_{D,j}(d_{ij}^2, \theta_{ij}, \alpha_{ij}) \quad (8)$$

$$P_{F,j}(\mathbf{x}_i, \mathbf{q}_j) = P_{F,j} - w_{F,j}(d_{ij}^2, \theta_{ij}, \alpha_{ij}) \quad (9)$$

donde $w_{D,j}$ y $w_{F,j}$ son funciones que disminuyen el valor de $P_{D,j}$ y $P_{F,j}$ con la distancia entre la celda i y el sensor j , d_{ij} , y los ángulos bajo los que se observa la celda i para la orientación del sensor j , θ_{ij} y α_{ij} (figura 3). La figura 4 muestra una gráfica de dichas funciones.

Si el sensor detecta fuego, entonces :

$$p(\mathbf{S}_j|h_i, \mathbf{q}_j) = P_{D,j}(\mathbf{x}_i, \mathbf{q}_j) \quad (10)$$

$$p(\mathbf{S}_j|\bar{h}_i, \mathbf{q}_j) = P_{F,j}(\mathbf{x}_i, \mathbf{q}_j) \quad (11)$$

En cambio, si el sensor no detecta fuego:

$$p(\mathbf{S}_j|h_i, \mathbf{q}_j) = 1 - P_{D,j}(\mathbf{x}_i, \mathbf{q}_j) \quad (12)$$

$$p(\mathbf{S}_j|\bar{h}_i, \mathbf{q}_j) = 1 - P_{F,j}(\mathbf{x}_i, \mathbf{q}_j) \quad (13)$$

La evaluación de (7) puede hacerse mediante muestreo a partir de la distribución $p(\mathbf{q}_j)$. El principal efecto de los errores en \mathbf{q}_j es una incertidumbre en los límites del campo de visión del sensor. Sin embargo, para el caso particular de los experimentos presentados en la sección 4, el UAV que porta el sensor de fuego es capaz de determinar su posición con gran precisión, por lo que no se tienen en cuenta.

3.2.2. Cámaras

Como resultado de los algoritmos de segmentación, los datos obtenidos a partir de las cámaras son imágenes binarias en las que los píxeles están clasificados como fuego o no fuego.

La función de medida de las cámaras considera el modelo de proyección de las cámaras (el llamado modelo pin-hole). Cada celda tiene una posición asociada \mathbf{x}_i , y el centro de dicha celda corresponderá con un píxel $\mathbf{m}_{j,i}$ en el plano de la imagen de la cámara j (si está en el campo de visión de dicha cámara). Si \mathbf{x}_i y $\mathbf{m}_{j,i}$ se expresan en coordenadas homogéneas, la posición del píxel viene dada por:

$$\mathbf{sm}_{j,i}^T = \mathbf{A}_j[\mathbf{R}_j \mathbf{t}_j] \mathbf{x}_i = \mathbf{f}(\mathbf{q}_j, \mathbf{x}_i) \quad (14)$$

donde \mathbf{A}_j es la matriz de calibración de la cámara j . Esta matriz se obtiene mediante un proceso de calibración. \mathbf{R}_j y \mathbf{t}_j son la rotación y traslación que localizan la cámara en el sistema de referencia global, y son calculadas en base a los sensores a bordo del UAV (GPS, IMU y encoders si se dispone de pan-tilt para las cámaras).

De nuevo, para calcular la verosimilitud $p(\mathbf{S}_j|h_i)$, (7) debe ser integrada para todos los posibles valores de \mathbf{q}_j . Y, además, esto debe ser hecho para todas las celdas del grid que se encuentran en el campo de visión de la cámara. Para determinar de forma precisa la posición del fuego, y a diferencia del caso del sensor de fuego, aquí es necesario tener en cuenta las incertidumbres en \mathbf{q}_j , pues las cámaras proporcionan mucha más información acerca de la localización del fuego.

La solución directa de (7) usando métodos de Monte-Carlo tiene un coste computacional elevado. Por ello, en lugar de resolver (7), las incertidumbres en \mathbf{q}_j se trasladan a incertidumbres en la posición del píxel correspondiente $\mathbf{m}_{j,i}$ a través de la función \mathbf{f} . Esta función es claramente no lineal. La propagación de las incertidumbres se realiza empleando la llamada Unscented Transform ([6]).

Por tanto, (7) se convierte en:

$$p(\mathbf{S}_j|h_i) = \sum_{\mathbf{m}} p(s_j|\mathbf{m}_{j,i}) p(\mathbf{m}_{j,i}) \quad (15)$$

El término $p(s_j|\mathbf{m}_{j,i})$ corresponde a (10) si el píxel $\mathbf{m}_{j,i}$ pertenece a una región segmentada como fuego, mientras que si el píxel fue clasificado como parte del fondo, entonces corresponde a (12). La misma aproximación se emplea para calcular $p(\mathbf{S}_j|\bar{h}_i)$.

Para las cámaras, las funciones $w_{D,j}$ y $w_{F,j}$ en (8) y (9) sólo dependen de la distancia de la celda a la cámara d_{ij} , y no de su orientación relativa.

3.3. OBTENCIÓN DE MEDIDAS DEL GRID

Mediante las ecuaciones anteriores se estima de forma recursiva el estado del grid usando los datos que los distintos UAVs de la flota van obteniendo.

Desde un punto de vista Bayesiano, el grid representa toda la información que puede obtenerse acerca de las posibles alarmas en el instante $k+1$. Sin embargo, en algunas aplicaciones se necesitan medidas más específicas. Por ejemplo, si un equipo de UAVs está rastreando una zona en busca de posibles incendios, el centro de control necesitaría una estimación de la posición de las alarmas potenciales, de modo que pueda planificar una nueva misión, enviando más UAVs a confirmar la alarma.

La posición de las alarmas se estima del siguiente modo. Cada T segundos se obtiene el conjunto de celdas del grid con una probabilidad superior a un determinado umbral. Una alarma se genera por cada conjunto R de celdas vecinas que superan dicho umbral (si la región supera un tamaño mínimo). La posición de la alarma se calcula como la media ponderada de las posiciones de las celdas, usando las probabilidades como pesos.

$$\boldsymbol{\mu}_R = \frac{\sum_{i \in R} \mathbf{x}_i p(h_i|\mathbf{S})}{\sum_{i \in R} p(h_i|\mathbf{S})} \quad (16)$$

También se obtiene una medida de la incertidumbre en esta posición estimada de los momentos de segundo orden de la región R .

4. EXPERIMENTOS

4.1. DESCRIPCIÓN DE LOS EXPERIMENTOS

Los experimentos con fuegos controlados del proyecto COMETS han tenido lugar en el aeródromo de Lousã (Portugal) en el mes de mayo de los años



Figura 6: Una vista de lugar de los experimentos desde el UAV Marvin.

2003, 2004 y 2005. El objetivo de los experimentos era la demostración final de la detección y monitorización automática de pequeños fuegos controlados por parte de varios UAVs heterogéneos, equipados con distintos sensores.

Se han empleado los siguientes UAVs en dichos experimentos: el helicóptero Marvin, el dirigible Karma y el helicóptero Heliv. Marvin es un helicóptero autónomo desarrollado por el Grupo de Sistemas en Tiempo Real y Robótica de la Technische Universität en Berlin ([10]). Karma es un dirigible autónomo de $18m^3$ desarrollado en el LAAS (Laboratoire d'Architecture et d'Analyse des Systèmes) [5]. Heliv es el resultado de la evolución de un helicóptero de radio-control convencional, al que le han sido añadidas sensores y capacidades de percepción, comunicación y control. La figura 5 muestra imágenes de todos ellos.

Como ya se ha comentado, los UAVs no son solo heterogéneos en cuanto a las plataformas (helicópteros y dirigibles) o su autonomía, sino también en los sensores que posee cada uno de ellos. Además de los receptores GPS giróscopos y sensores de medida inercial para la navegación, los UAVs cargan sensores para la percepción del entorno. Los sensores considerados en los experimentos son cámaras (visuales e infrarrojas) y un detector de fuego especializado (descritos en la sección 3.2).

Los helicópteros, además, poseen unidades pan-and-tilt motorizadas que permiten orientar las cámaras independientemente del cuerpo del vehículo. Estas unidades poseen codificadores ópticos para medir los ángulos. Además, el dirigible Karma tiene tamaño suficiente para cargar un par estéreo con una línea de base de unos 10 m., que puede ser usada para obtener mapas de elevación [5].

El escenario en el que ocurre la misión es un cuadrado de unos 400 metros de lado. La figura 6 muestra el escenario de los experimentos visto desde la cámara de abordo de unos de los vehículos involucrados, el helicóptero autónomo Marvin.

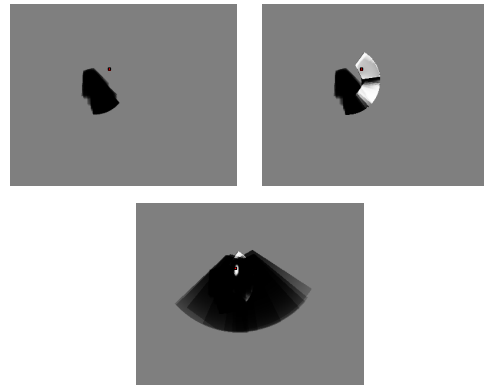


Figura 7: El estado del grid en tres instantes de la misión. El cuadrado marca la posición real del fuego.

4.2. ESQUEMA DE LA MISIÓN Y RESULTADOS

Durante los experimentos, se probaron en años sucesivos distintos aspectos relacionados con la coordinación y control de varios UAVs. En los experimentos de 2005 se incluyó una demostración de una misión general de detección, confirmación y monitorización de incendios. Dicha misión se describe en lo que sigue.

Durante la misión, se provocó un pequeño fuego controlado usando balas de paja. Para validar los resultados, la posición del fuego es registrada usando un receptor GPS.

El centro de control planifica una primera fase en la que un UAV es enviado a sobrevolar una zona, buscando posibles alarmas. En este caso, Marvin recibe órdenes para el despegue, una serie de puntos de paso (calculados para cubrir una zona determinada) y la orden de realizar tareas de percepción para la detección.

La figura 7 muestra la evolución del grid en varias fases del experimento (colores oscuros indican baja probabilidad). La dos primeras imágenes muestran como evoluciona el estado del grid mientras Marvin ejecuta su misión inicial. En la primera, Marvin ha volado sobre una zona donde no hay fuego, usando únicamente el sensor de fuego que lleva a bordo. La segunda imagen muestra como Marvin genera dos zonas de alta probabilidad en el grid, una debido a una falsa alarma y otra debida al fuego real.

Por tanto, se generaron dos alarmas. El centro de control reacciona, y se re-planifica la misión actual. Heliv es enviado para la confirmación y localización precisa, mientras que Marvin es enviado a una posición segura y mantenido en vuelo estacionario. Heliv usa su cámara IR en la zona de las



Figura 5: Heliv, Marvin y Karma en acción.

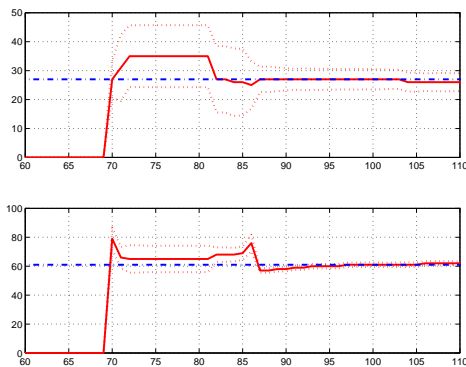


Figura 8: Posición estimada de una de las zonas de alta probabilidad (rojo) y posición real del fuego (azul). Se muestra la varianza de la posición estimada.

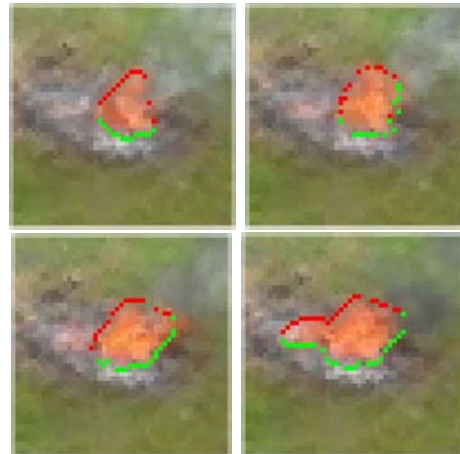


Figura 9: Medidas del fuego obtenidas.

alarmas. El sistema de percepción combina la estimación anterior con los nuevos datos de Heliv. La tercera imagen de la figura 7 muestra como tras cierto tiempo, cuando las imágenes IR de Heliv y las lecturas del sensor de fuego son integrados, la región de alta probabilidad se reduce a una zona más pequeña, que incluye la posición real del fuego. Del mismo modo, la zona correspondiente a la falsa alarma es descartada.

Además, al combinar los datos a partir de las cámaras de Heliv, la posición del fuego se determina con mucha mayor precisión. La figura 8 muestra la evolución de la posición del fuego estimada mediante (16) y comparada con la posición real. También muestra una estimación de la incertidumbre en dicha posición estimada.

Una vez confirmada la alarma, el centro de control de nuevo produce nuevas misiones para los UAVs. Karma entra en acción. Se le ordena despegar y tomar imágenes desde gran altitud para supervisar la zona. La figura 10 muestra un mosaico obtenido con las imágenes capturadas por Karma. Al mismo tiempo, Heliv y Marvin son enviados a tomar de forma sincronizada imágenes del incendio para la monitorización del mismo (ver figura 9).

Tanto la sincronización como la fusión de datos

tienen lugar de forma distribuida entre los UAVs. Una vez completada la misión, los 3 UAVs reciben la orden de regresar y aterrizar.

Todo lo descrito en este apartado sucede de forma autónoma. La intervención humana se centra en la teleoperación de uno de los helicópteros y en las fase de replanificación, que son confirmadas por un operador. El procesamiento y obtención de datos es totalmente automático.

5. CONCLUSIONES

El proyecto COMETS, realizado con la coordinación científica y técnica de la Universidad de Sevilla, alcanzó todos los objetivos de la propuesta y fue evaluado muy positivamente por los revisores de la Comunidad Europea, quienes pusieron de manifiesto que el proyecto había contribuido significativamente en robótica, sistemas empotrados, aeronáutica, seguridad y medio ambiente, que había generado una perspectiva más madura sobre el diseño de sistemas autónomos basados en la cooperación de objetos, y que había generado nuevos productos y sistemas en arquitecturas de control, tecnología de UAV de bajo coste, percepción, teleoperación y comunicaciones.

El trabajo ha dado lugar al primer sistema para la



Figura 10: Mosaico obtenido a partir de las imágenes de Karma.

coordinación y control en tiempo real de múltiples vehículos aéreos autónomos heterogéneos del que se tiene conocimiento que haya sido implantado y validado con los UAVs. Se trata también del primer sistema en el mundo que ha sido aplicado a la detección y monitorización de incendios forestales.

Agradecimientos

Este trabajo ha sido financiado por los proyectos COMETS, de la Comisión Europea (Contract No. IST-2001-34304), y AEROSSENS (Dirección General de Investigación DPI2005-02293). Los autores agradecen la participación del resto de miembros del proyecto COMETS en los experimentos descritos.

Referencias

- [1] Markus Eisl and Mehrdad Khalili. ARC - airborne minefield area reduction. In J. Cornelis H. Sahli, A.M. Bottoms, editor, *Proceedings of the EUDEM2-SCOT-2003 Conference*, 2003.
- [2] J. Gancet, G. Hattenberger, R. Alami, and S. Lacroix. Task planning and control for a multi-uav system: architecture and algorithms. In *Proceedings of the IEEE/RSJ IROS 2005*, pages 1017–1022, August 2005.
- [3] J. Gancet and S. Lacroix. Embedding heterogeneous levels of decisional autonomy in multi-robots systems. In *Proceedings of the 7th International symposium on Distributed Autonomous Robot Systems*, Toulouse, France, June 2004.
- [4] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems Magazine*, pages 34–44, December 2000.
- [5] E. Hygounenc, I-K. Jung, P. Soueres, and S. Lacroix. The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping. *The International Journal of Robotics Research*, 23(4-5):473–511, 2004.
- [6] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of the 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.
- [7] J.R. Martínez-de-Dios and A. Ollero. A multiresolution threshold selection method based on training. *Lecture Notes in Computer Science*, 3211:90–97, 2004.
- [8] A. Ollero, J. Alcázar, F. Cuesta, F. López-Pichaco, and C. Nogales. Helicopter teleoperation for aerial monitoring in the COMETS multi-UAV system. In *3rd IARP Workshop on Service, Assistive and Personal Robots*, 2003.
- [9] W. Phillips, M. Shah, and N. da Vitoria Lobo. Flame recognition in video. *Pattern Recogn. Lett.*, 23(1-3):319–327, 2002.
- [10] V. Remuss, M. Musial, and G. Hommel. Marvin - an autonomous flying robot-bases on mass market. In *International Conference on Intelligent Robots and Systems, IROS. Proceedings of the Workshop WS6 Aerial Robotics*, pages 23–28. IEEE/RSJ, 2002.
- [11] G.S. Sukhatme, J. Montgomery, and R.T. Vaughan. *Robot Teams: From Diversity to Polymorphism*, chapter Experiments with Aerial-Ground Robots. AK Peters, 2001.
- [12] R. Vidal, S. Sastry, J. Kim, O. Shakernia, and D. Shim. The berkeley aerial robot project (bear). In *Proceeding of the International Conference on Intelligent Robots and Systems, IROS*, pages 1–10. IEEE/RSJ, 2002.

ACTUALIZACIÓN DE MAPAS DE CARACTERÍSTICAS BASADA EN LA POSICIÓN RELATIVA DE LAS ESQUINAS

De la Casa Cárdenas, J., Gómez Ortega, J.
Grupo de Robótica, Automática y Visión por Computador
Escuela Politécnica Superior de Jaén. Universidad de Jaén
Campus de las Lagunillas. 23071. Jaén
{casacar, juango}@ujaen.es

Resumen

Este artículo muestra una nueva técnica de actualización de mapas de características (segmentos) a partir de los datos proporcionados por un sensor láser montado sobre un robot móvil autónomo. Básicamente, se trata de comparar la posición relativa de todas las esquinas del mapa local con las del mapa global. La ventaja fundamental se halla en la invariabilidad de los segmentos que unen las esquinas en un mapa y en otro. Si en el mapa local y en el mapa global se están observando un mismo grupo de esquinas, la posición relativa de las mismas será igual en los dos mapas. Así, los segmentos que las unen tendrán la misma longitud. Este hecho se utiliza para calcular el cambio de orientación y desplazamiento de un mapa con respecto a otro para así determinar la transformación a realizar.

Palabras Clave: mapas de características, láser, SLAM.

1 INTRODUCCIÓN

Un robot móvil autónomo que navega en entornos interiores desconocidos debe ser capaz de conocer su posición. Esta tarea, junto con la construcción del mapa del entorno se engloba en un concepto denominado SLAM (Simultaneous Localization and Mapping). Estas dos acciones suelen ir unidas, puesto que para localizarse dentro de un espacio se debe conocer una representación del entorno (ya sea por la construcción en tiempo real del mismo o por su conocimiento *a priori*) y para construir un mapa es necesario conocer la posición del robot. Este entorno puede ser estático o, incluso, dinámico [5].

La construcción del mapa suele basarse en celdillas de ocupación o en la extracción de las características del mismo; habitualmente, segmentos y polilíneas. No obstante, la representación de las características del entorno implica un menor coste computacional y

está siendo el método de representación más utilizado en los últimos años.

En cuanto al proceso de construcción, consiste, básicamente, en ir actualizando un mapa global a partir de los mapas locales que genera el robot. Estos mapas locales se van obteniendo en cada una de las posiciones en las que el robot toma muestras con su sistema sensorial.

El problema consiste en la superposición de los mismos. Hay varios criterios para realizar la actualización. Por ejemplo, a partir de los segmentos extraídos [3], segmentos y esquinas [2], polilíneas [6] incluso utilizando técnicas basadas en Biología Computacional [4].

En este trabajo se propone un procedimiento basado en la comparación del grafo obtenido a partir de las esquinas presentes en los mapas de entornos interiores parcialmente estructurados. Este grafo se construye uniendo todas las esquinas del mapa.

La ventaja de elegir puntos del plano (esquinas) y no segmentos para la comparación del mapa local y global es el menor grado de incertidumbre que existe al comparar puntos en lugar de segmentos. Una pared vista desde dos posiciones distintas puede no apreciarse de la misma forma (por ejemplo, se puede observar parcialmente en una posición y completa en otra); eso puede suponer un grado mayor de dificultad a la hora de la comparación de segmentos. Sin embargo, si el mismo punto se observa desde dos posiciones diferentes, esta característica no permite una observación total o parcial.

No obstante, los elementos que se comparan no son los puntos que determinan las esquinas, sino las uniones del grafo formado a partir de las mismas. Así, una unión aparecerá en los dos grafos si y solo si en el mapa están representadas las esquinas. A partir de sus características geométricas (longitud y orientación) es posible determinar el valor de la traslación y la rotación del robot.

Este procedimiento, aunque diseñado para actuar independientemente, puede utilizarse como complemento a otros para aumentar el nivel de confianza y disminuir el grado de incertidumbre.

Este artículo se estructura de la siguiente forma: en la sección 2 se presenta el proceso de construcción del mapa de características, definiendo las fases del mismo y explicando el método que se propone en este trabajo. La sección 3 muestra las pruebas realizadas, tanto de simulación como reales, finalizando, en la sección 4, con las conclusiones obtenidas.

2 CONSTRUCCIÓN DEL MAPA DE CARACTERÍSTICAS

Antes de explicar el algoritmo de actualización del mapa global, es importante exponer los distintos pasos que se llevan a cabo previamente; éstos incluyen la lectura de los datos por parte del sensor láser, la segmentación, el filtrado de los segmentos obtenidos y, por último, la actualización del mapa.

2.1 OBTENCIÓN DE DATOS

El equipo utilizado es un escáner láser programable (PLS) de la empresa SICK montado en un robot móvil autónomo ATRV-Mini de iRobot tal y como aparece en la figura 1. El láser tiene un alcance de 50m., realizando un barrido de 180° con una resolución de hasta 0,5°. Actualmente, está programado para una resolución de 1° con lo que en cada lectura se obtienen 180 medidas.



Figura 1: ATRV-Mini

2.2 SEGMENTACIÓN

Una vez que el láser ha tomado una lectura del entorno, se realiza el proceso de segmentación. Esta tarea consiste en obtener una serie de segmentos que se corresponden con las paredes del espacio que está recorriendo el robot.

De entre los diferentes métodos que hay para la extracción de rectas a partir de una lectura del sensor láser [7][8], se ha optado por la transformada de Hough, utilizando el método propuesto en [1] para la elección del umbral de votos.

Posteriormente a la obtención de las rectas, se definen los segmentos. En primer lugar se toman los puntos cercanos a cada recta, fijando un umbral de cercanía de los unos con respecto a las otras y, posteriormente, se agrupan los puntos más cercanos formando, finalmente, los segmentos; no obstante, se eliminan aquéllos cuya longitud no sea superior a una mínima predefinida. Estos segmentos se parametrizan (coordenadas de los extremos, longitud, orientación) y se procede a un filtrado de los mismos para eliminar segmentos que no aportan información del entorno.

2.2.1 Filtrado de segmentos

Este filtrado tiene un doble objetivo. Por un lado, eliminar segmentos que no ofrezcan información y por otro, determinar las esquinas que se hayan podido observar desde la posición actual del robot. Se asume que el entorno está compuesto por paredes rectas que forman ángulos de 90°, aproximadamente; por tanto, se considerará que hay una esquina en el mapa cuando dos segmentos formen un ángulo cercano a 90°. Para eliminar segmentos se tienen en cuenta la longitud y la orientación relativa de los dos que se estudian. La casuística que se utiliza es la siguiente:

- Si un segmento corta a otro y forman un ángulo distinto de 90° (con cierto margen de error) se elimina el más pequeño puesto que proporciona menos información del entorno. En la figura 2, los segmentos mostrados en línea discontinua se eliminarían del mapa.

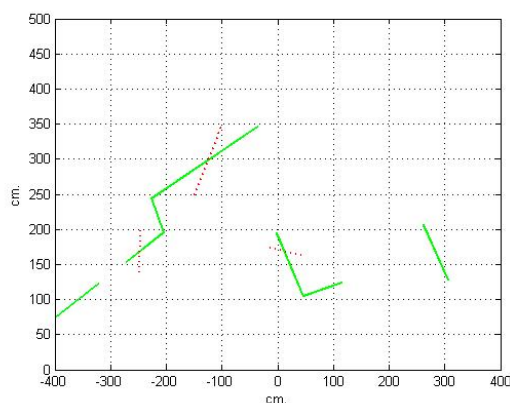


Figura 2: Filtrado de segmentos

- Si dos segmentos se cortan, formando un ángulo de 90° (asumiendo una pequeña tolerancia) se considera que forman una

esquina y esa esquina se sitúa en el punto de corte de ambos segmentos. Las coordenadas de los extremos más cercanos de los segmentos se modifican para hacerlas coincidir con el punto de corte. La figura 3 muestra esta situación; el segmento de línea discontinua se añade al conjunto de características así como el punto de unión, como una esquina.

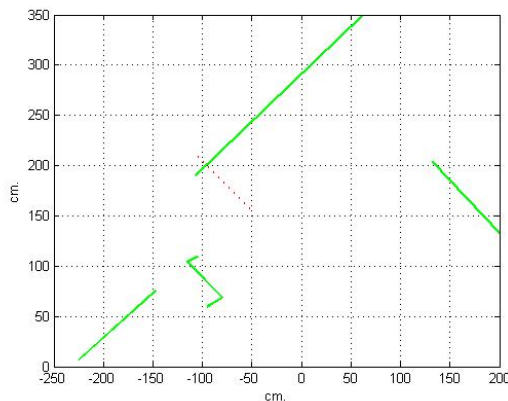


Figura 3: Filtrado de segmentos

- Otra posibilidad es que dos segmentos no se crucen pero sí lo hagan las rectas a las que pertenecen, estando este punto de corte cerca de algún extremo de los dos segmentos. En ese caso, se considera que también hay una esquina y se cambian los extremos de los segmentos para hacerlos coincidentes con el punto de corte de las rectas. En la figura 4, se muestra la nueva esquina detectada en el mapa.

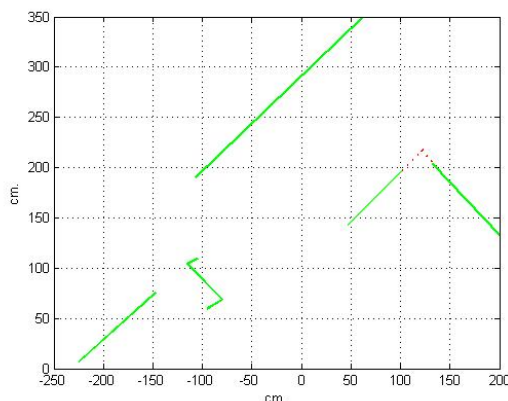


Figura 4: Filtrado de segmentos

En definitiva, se pretende obtener los segmentos que mejor definan el entorno y la posición de las esquinas. Esta es la información que necesita este método para actualizar el mapa.

2.3 ACTUALIZACIÓN DEL MAPA GLOBAL

Una vez que se ha obtenido el mapa local, y si no es la primera lectura que se realiza, se procede a la actualización del mapa global.

El primer paso es calcular el grafo de las esquinas del mapa local, tal y como aparece en la figura 5. Estos segmentos serán los que se comparen con los del mapa global.

Si sólo se ha detectado una esquina, ese mapa local no es válido al no poder construir los segmentos; el robot deberá moverse a otra posición del entorno.

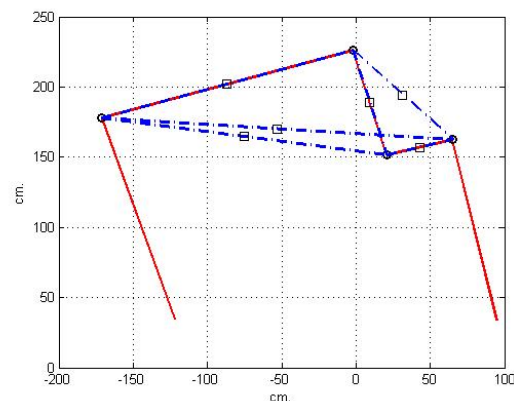


Figura 5: Construcción del grafo a partir de las esquinas del mapa

El algoritmo se muestra a continuación:

Paso 1. Se compara la longitud de todas las uniones del grafo del mapa local con las del mapa global y se eligen aquellas cuya longitud sea similar; se considera un grado de incertidumbre determinado a partir de los datos del error del sensor láser.

Paso 2. De entre los segmentos elegidos, se calcula la variación de orientación de cada uno de ellos y sus similares.

Paso 3. Al final, se obtienen un conjunto de segmentos (idealmente, sólo un segmento) similar a cada uno de los que aparecen en el mapa local. Se pueden utilizar métodos estadísticos para calcular la orientación. En este caso, se ha determinado qué valores aparecen con más frecuencia y, entre ellos, se ha calculado la media.

Con esa información se actualiza el mapa global, transformando sus coordenadas según la matriz de rotación (1).

$$M_{rot} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix} \quad (1)$$

A continuación, y a partir de la diferencia en las coordenadas de los segmentos similares, se calcula el desplazamiento que se ha realizado.

3 PRUEBAS REALIZADAS

Se han realizado dos grupos de pruebas diferentes. El primero se basa en simulaciones de varios casos posibles en los que se observa cómo se comporta el algoritmo en cada ocasión. El segundo corresponde a un experimento real con el robot que dispone el grupo de investigación.

3.1 SIMULACIONES

Se ha efectuado un conjunto de pruebas a partir de una serie de mapas ficticios que se han ido modificando para incluir distintas dificultades.

El mapa inicial (en línea continua de color rojo), indicando las uniones del grafo (línea discontinua de color azul), se muestra en la figura 6. La tolerancia que se ha fijado cuando se compara la longitud de los segmentos ha sido de 10cm. y la variación máxima permitida en la orientación ha sido de 15°.

El objetivo es analizar el comportamiento del procedimiento propuesto ante diferentes situaciones que se pueden encontrar en mapas reales. En los resultados que se ofrecen, aparecen los valores de rotación que obtiene el algoritmo junto con la frecuencia de aparición en valor absoluto y porcentual; es decir, sobre el total de posibles orientaciones, el porcentaje en el que aparece cada una.

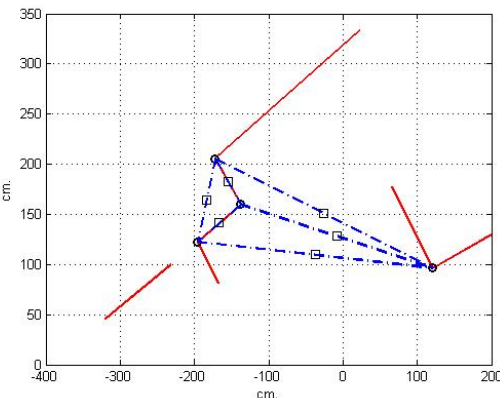


Figura 6: Mapa inicial

3.1.1 Simulación 1

En la primera prueba se ha realizado una traslación de (10,0) y una rotación de 10° a partir del mapa inicial. El resultado, aparece en la figura 7.

Como se observa, aparecen las mismas esquinas y, por tanto, el grafo es idéntico. Este es el caso más favorable puesto que todas las uniones aparecen en los dos mapas. Al aplicar el algoritmo, el resultado es el que aparece en la tabla 1.

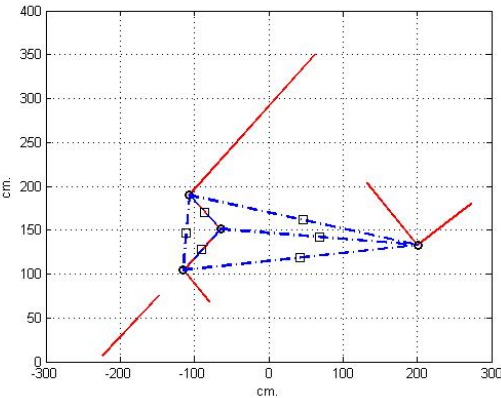


Figura 7: Simulación 1

Tabla 1: Resultado de la simulación 1

Ángulo	Frecuencia	Porcentaje
10°	6	85.71%
-6°	1	14.29%

3.1.2 Simulación 2

A partir del mapa anterior, se ha eliminado una esquina, tal y como aparece en la figura 8. Los colores y tipo de línea son iguales que en apartado anterior.

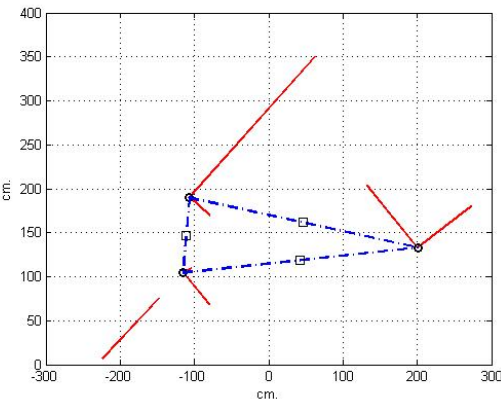


Figura 8: Simulación 2

El resultado se muestra la tabla 2.

Tabla 2: Resultado de la simulación 2

Ángulo	Frecuencia	Porcentaje
10°	3	75%
-6°	1	25%

3.1.3 Simulación 3

En este caso se ha añadido una esquina más al mapa original. De nuevo, el número de uniones del grafo es distinto, tal y como se aprecia en la figura 9.

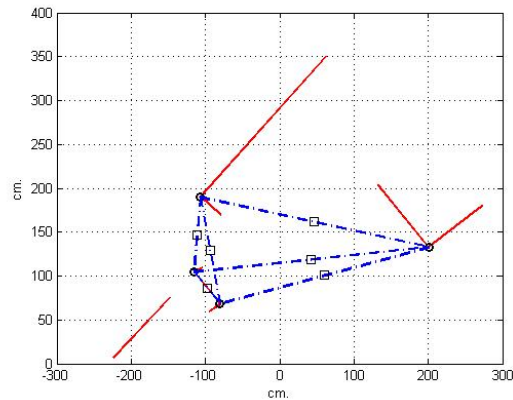


Figura 9: Simulación 3

El resultado aparece en la tabla 3.

Tabla 3: Resultado de la simulación

Ángulo	Frecuencia	Porcentaje
10°	3	60%
-6°	1	20%
7°	1	20%

En las tres simulaciones, el algoritmo ofrece la solución correcta.

3.2 EXPERIMENTO REAL

Una vez que se han mostrado diferentes simulaciones, se presenta ahora un caso real, en el que se utilizan datos obtenidos por el robot ATRV-Mini en el laboratorio del grupo de investigación.

3.2.1 Cálculo de la variación de orientación

Se han tomado muestras de una zona del laboratorio desde tres posiciones distintas. Estos tres mapas se muestran en las figuras 10, 11 y 12. Además, se ha leído la odometría del robot, proporcionando los datos que aparecen en la tabla 4.

Tabla 4: Odometría en el experimento real

Mapa 01:		
x = 0.0cm.	y = 0.0cm.	$\phi = 0.0^\circ$
Mapa 02:		
x = 0.0cm.	y = 0.0cm.	$\phi = -11.06^\circ$
Mapa 03:		
x = 6.43cm.	y = 32.98cm.	$\phi = -11.06^\circ$

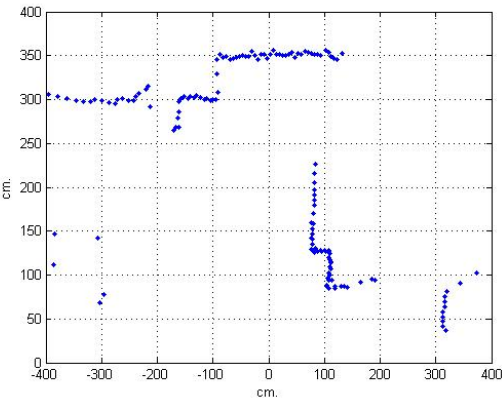


Figura 10: Mapa 01

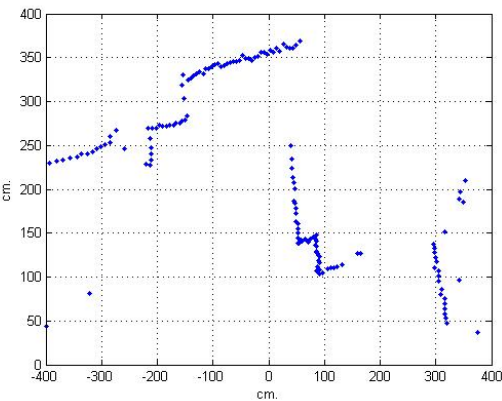


Figura 11: Mapa 02

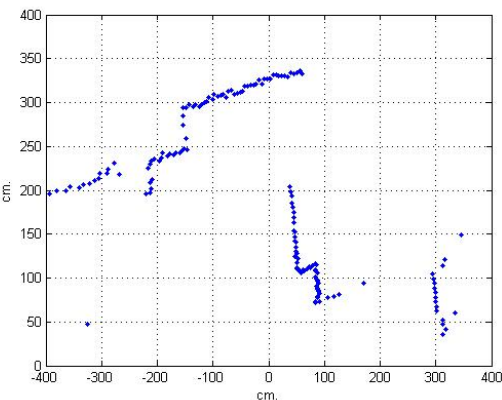


Figura 12: Mapa 03

Una vez realizado el proceso de segmentación y construido el grafo de las esquinas, los resultados se muestran en las figuras 13, 14 y 15. Las características están en línea continua y de color rojo, mientras que el grafo, en línea discontinua y color azul. También se indican las esquinas.

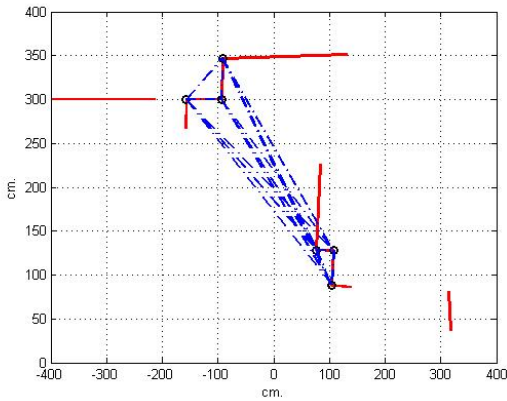


Figura 13: Grafo del Mapa 01

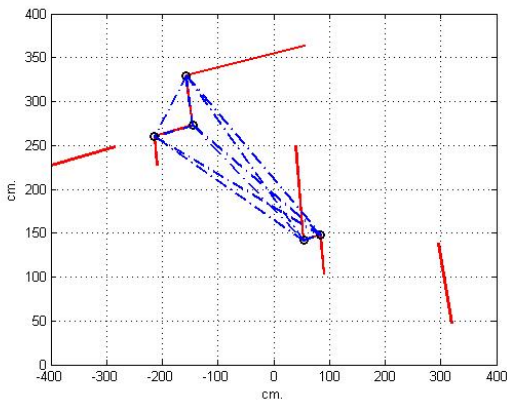


Figura 14: Grafo del Mapa 02

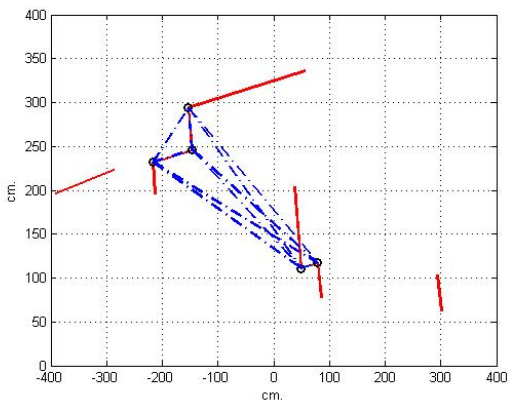


Figura 15: Grafo de Mapa 03

Para observar mejor la bondad del método, se ha aplicado el mismo a las tres posibles combinaciones de los tres mapas. Puesto que el desplazamiento ha sido pequeño, pueden compararse los resultados con la lectura de la odometría. Al final, se actualiza el mapa 01 con las características observadas en los mapas 02 y 03.

En la tabla 5 aparecen los resultados obtenidos al aplicar el algoritmo al mapa 01 y mapa 02,

considerando el mapa 01 como global y el mapa 02, local.

Tabla 5: Mapa 01 vs. Mapa 02

Ángulo	Frecuencia	Porcentaje
12°	2	28.57%
-6°	1	14.29%
5°	1	14.29%
11°	2	28.57%
13°	1	14.29%

El valor máximo se obtiene para las variaciones de orientación de 12° y 11°. Haciendo una media, obtenemos un valor final de 11.5°. Según la odometría, el valor de la orientación debería ser de 11.06°.

En el caso del mapa 01 y el mapa 03, los resultados se muestran en la tabla 6.

Tabla 6: Mapa 01 vs. Mapa 03

Ángulo	Frecuencia	Porcentaje
11°	3	23.08%
9°	1	7.69%
12°	3	23.08%
10°	4	30.77%
-2°	1	7.69%
-1°	1	7.69%

El máximo se obtiene para el valor 10°.

Por último, al analizar el mapa 02 y el mapa 03, según se observa en la tabla 7, el resultado es 0°.

Tabla 7: Mapa 02 vs. Mapa 03

Ángulo	Frecuencia	Porcentaje
-3°	1	16.67%
-1°	2	33.33%
0°	3	50.00%

3.2.2 Actualización del mapa

Para actualizar el mapa 01 con la información del mapa 02, se calcula la rotación, como se ha hecho anteriormente y se transforman las coordenadas de los segmentos de acuerdo con la correspondiente matriz de transformación (1). En la figura 16, en color azul aparece el mapa 01 y en color rojo el mapa 02, una vez aplicada la matriz de rotación

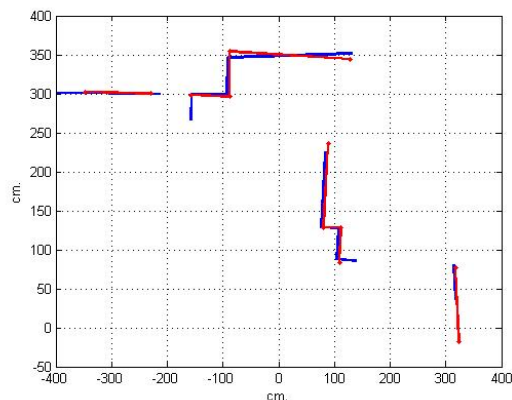


Figura 16: Mapa 01 y Mapa 02 aplicando la rotación

Posteriormente, se toman los segmentos que unen las esquinas similares en un mapa y otro y se determina la posible traslación que pueda haber. En este caso, al hacer la media, resulta una traslación de (4.71, 1.03). El resultado final aparece en la figura 17.

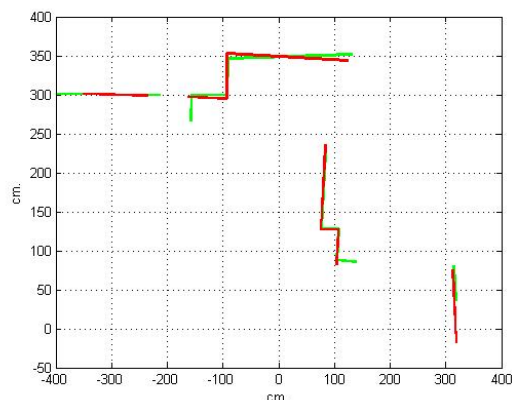


Figura 17: Actualización del Mapa 02 sobre Mapa 01

Para la actualización del mapa 03 sobre el mapa 01, la variación de orientación es de 10.00° . Al aplicar la rotación queda la situación mostrada en la figura 18.

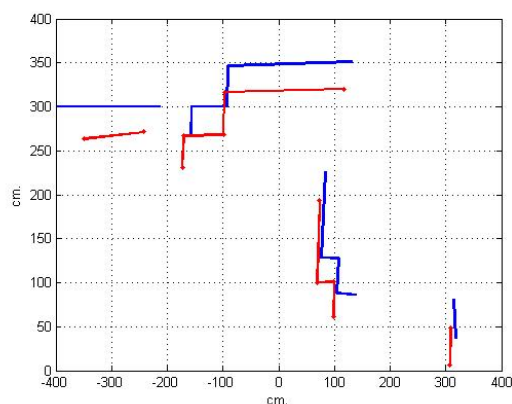


Figura 18: Mapa 01 y Mapa 03 aplicando la rotación

La traslación resultante al aplicar el mismo método que en el caso anterior resulta (-6.81, -30.46). Al final, queda como aparece en la figura 19.

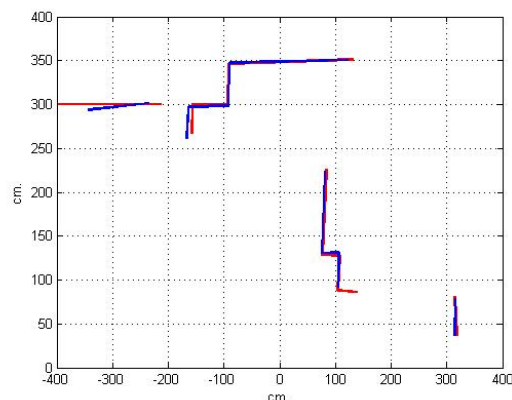


Figura 19: Actualización del Mapa 03 sobre Mapa 01

Si se actualiza el mapa 01 con la información del mapa 02 y mapa 03, la situación queda como se muestra en la figura 20.

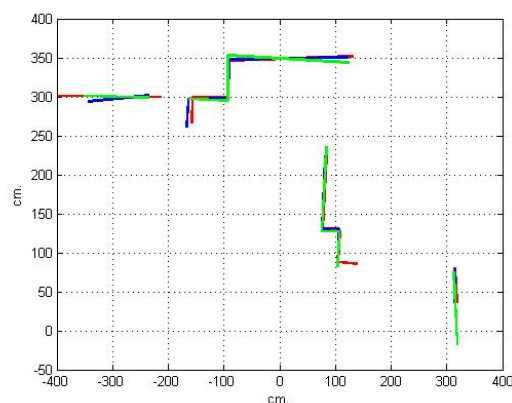


Figura 20: Mapa 01, Mapa 02 y Mapa 03

4 CONCLUSIONES

Se ha mostrado un método para actualizar mapas de entorno a partir de los datos proporcionados por un sensor láser. Aunque se han utilizado los datos procedentes de un sensor láser, las características pueden obtenerse a partir de cualquier sensor. Sólo es necesario disponer de los segmentos que definan esas características y las esquinas observadas.

Al realizar la actualización a partir de la posición relativa de las esquinas, se evita comparar todas las características del entorno.

Además, las uniones del grafo, al estar formadas a partir de las esquinas, tienen la misma longitud en los dos mapas, evitando así la incertidumbre en cuanto a determinar la correspondencia entre dos segmentos.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el proyecto DPI2004-04458 del Plan Nacional de Investigación Científica, Desarrollo en Innovación Tecnológica 2004/2007.

Referencias

- [1] De la Casa Cárdenas, J., Gómez Ortega, J., (2005) Método heurístico para la extracción dinámica de rectas utilizando la transformada de Hough, XXVI Jornadas de Automática.
- [2] Diosi, A., Kleeman, L., (2004) Advanced sonar and laser range finder fusion for simultaneous localization and mapping, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [3] González, J. et al., (1994) Map building for a mobile robot equipped with a 2D laser rangefinder, IEEE International Conference Robotics and Automation (ICRA'94).
- [4] González-Buesa, C., Campos, J., (2004) Solving the mobile robot localization problem using string matching algorithms, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [5] Hähnel, D. et al., (2003) Map building with mobile robots in dynamics environments, IEEE International Conference on Robotics and Automation (ICRA'03).
- [6] Latecki, L.J. et al. (2004) Building polygonal maps from laser range data, ECAI Int. Cognitive Robotics Workshop.
- [7] Nguyen, V. et al., (2005) A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05).
- [8] Sack, D., Burgard, W., (2004) A comparison of methods for line extraction from range data, Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV).

Ejemplo de Integración de Redes de Comunicación en Entornos Industriales

José Manuel Moreno, Manuel Berenguel, Francisco Rodríguez, José Luís Guzmán
 Dpto. Lenguajes y Computación. Universidad de Almería.
 Carretera de Sacramento s/n 04120 Cañada de San Urbano
jmorenopuertas@gmail.com, beren@ual.es, frrodrig@ual.es, joguzman@ual.es

Resumen

En la automatización industrial, los buses de comunicación cada día juegan un papel más importante, permitiendo a diferentes dispositivos (sensores, actuadores, autómatas, terminales de supervisión) interactuar entre sí dentro de una arquitectura distribuida. Actualmente el principal inconveniente de este tipo de arquitecturas es la gran cantidad de dispositivos, fabricantes, protocolos, etc. Para solventar este problema es necesaria la integración de todos los sistemas dotándolos de una mayor flexibilidad. En este trabajo se describe el desarrollo de una arquitectura para la integración de un sistema de comunicaciones industriales tomando como referencia, los distintos niveles de abstracción sobre integración y producción de acuerdo a la filosofía "Computer Integrated Manufacturing" (CIM), barajando distintas alternativas en función de la tecnología de comunicación utilizada para cada nivel.

Palabras Clave: Buses de Campo, Integración, Redes de Comunicación.

1 INTRODUCCIÓN

En la industria coexisten equipos y dispositivos dedicados al control de procesos. Entre estos sistemas se encuentran los autómatas programables, computadores de diseño y gestión, sensores y actuadores, etc. El desarrollo de las redes industriales ha proporcionado un modelo para unir estos dispositivos aumentando el rendimiento y aportando nuevas características a los sistemas. Esta integración total se conoce como CIM (Computer Integrated Manufacturing)[4].

El objetivo principal es la comunicación totalmente integrada dotando a los sistemas de gran flexibilidad permitiendo integrar dispositivos de características diferentes mediante interfaces estandarizadas[4].

Aunque la utilización de las redes industriales debe ser estudiada y evaluada en terminos de coste, ofrece ventajas como:

- Visualización y Supervisión de todo el proceso.
- Toma de datos del proceso de forma más rápida.
- Mejora del rendimiento global de proceso.
- Posibilidad de intercambio de datos entre sectores del proceso y departamentos.
- Programación a distancia sin necesidad de estar a pie de fábrica.

En la implementación de las redes de comunicación en la industria de procesos se deben tener en cuenta ciertos requisitos:

- Demanda de alta seguridad
- Necesidad o posibilidad de funcionar en áreas peligrosas.
- Así como que sean sistemas abiertos y extensibles.

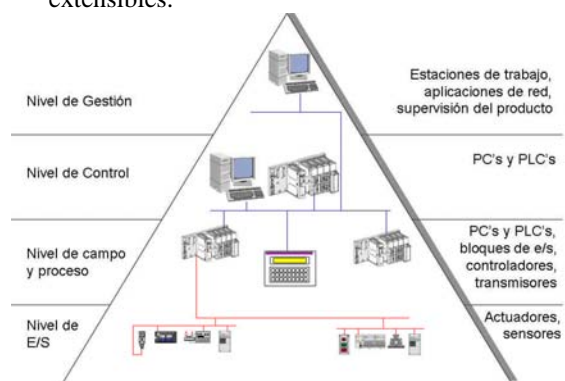


Figura1. Pirámide de Automatización

Continuando con el planteamiento de la filosofía CIM, este trabajo presenta un ejemplo de arquitectura para la integración de los niveles de automatización que forman un sistema, mostrados en la figura 1, así como la integración de sistemas sobre TCP-IP [6]. Dicha arquitectura ha sido desarrollada en base a las tecnologías de comunicación actuales ([1], [2], [3], [7], [10], [11], [12], [13], [14]).

La estructura del artículo se presenta como sigue: en la sección 2 se establecen los criterios y el planteamiento seguido en el trabajo, en la sección siguiente se exponen las tecnologías comunicación consideradas para el desarrollo de las interfaces de comunicación, en la sección 4 se desarrollan los niveles de comunicación para integrar una célula de

fabricación flexible [8], [15], y finalmente se exponen las conclusiones obtenidas.

2 DISTRIBUCIÓN DE REDES DE COMUNICACIÓN

Partiendo de la pirámide de automatización mostrada en la figura 1, se puede obtener una pirámide de comunicación tal como se observa en la figura 2.



Figura 2. Niveles de Comunicación

La pirámide de comunicación se encuentra constituida por los siguientes niveles:

- Nivel de Campo que comunica el nivel de entradas y salidas con el nivel de campo y proceso. Está formado principalmente por redes de área de bus de campo (FAN) [10].
- Nivel de Célula que comunica el nivel de campo y proceso con el nivel de control. Constituido por redes de área de bus de campo (FAN) y área local (LAN) [6].
- Nivel de Planta donde se comienzan a gestionar grandes volúmenes de datos y es el enlace entre el nivel de control y el nivel de gestión. Se encuentra formado por redes de área local y por redes de área amplia (WAN) [6].
- Nivel de Factoría donde la comunicación se establece entre grandes sistemas, mediante redes de alta disponibilidad como Ethernet. Esta formado por redes de área global (GAN) [6].

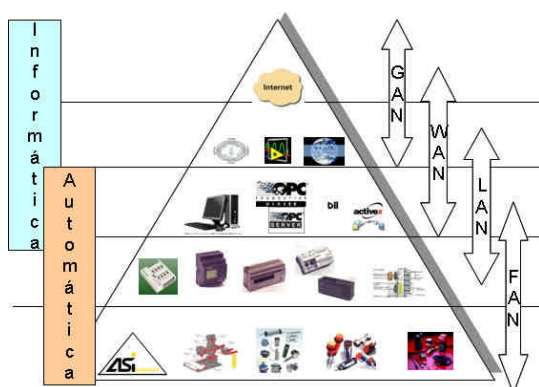


Figura 3. Distribución de redes industriales

Para el desarrollo concreto de este trabajo y continuando con el criterio establecido en los niveles de comunicación se han considerado dos niveles de interacción, un primer nivel que se encuentra formada por los niveles más bajos de la pirámide de automatización y que se encarga de la interacción con los dispositivos y un segundo nivel que se encarga de la gestión de datos abstrayéndose de los dispositivos permitiendo de esta manera flexibilizar los sistemas.

3 NIVELES DE COMUNICACIÓN

Para ir mostrando las distintas tecnologías utilizadas en cada nivel se han desarrollado diversas implementaciones partiendo de un problema recogido en [16], que consiste en el procedimiento para el tratamiento de superficies, con el fin de hacerlas resistentes a la oxidación. En la figura 4 se ilustra el proceso a automatizar.

El sistema consta de tres baños:

- Uno para el desengrasado de las piezas.
- Otro para el aclarado de las piezas.
- Un tercero donde se les dará el baño electrolítico.

La grúa introducirá la jaula portadora de las piezas a tratar en cada uno de los baños, comenzado por el de desengrasado, a continuación en el de aclarado y por último le dará el baño electrolítico; en este último, la grúa debe permanecer un tiempo determinado para conseguir una uniformidad en la superficie de las piezas tratadas.

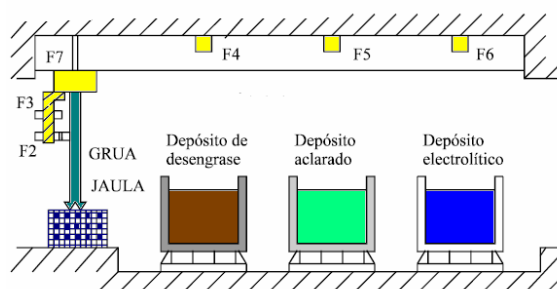


Figura 4. Esquema de tratamiento electrolítico

3.1 NIVEL DE AUTOMÁTICA

Esta área abarca los niveles de comunicación de célula y nivel de campo. También se puede considerar el nivel de entrada / salida como el nivel de comunicación entre los sensores, actuadores y el proceso propiamente dicho.

Para ilustrar el nivel de entrada y salida existen una amplia variedad de dispositivos en el mercado en este

caso particular se han utilizado tarjetas del fabricante ICPDAS [17] y una simulación de entradas a autómatas programables como se ve en la figura 5.

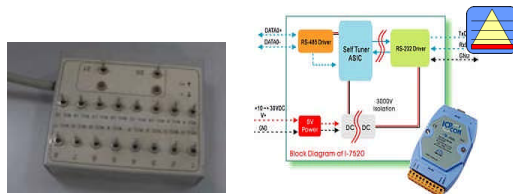


Figura 5. Dispositivos nivel E/S

Por lo general suelen ser dispositivos que utilizan transistores, y relés, para la lectura y escritura del proceso.

Los otros dos niveles que se encuentran en esta área, están formados principalmente por buses de campo. Dentro de la clasificación según sus prestaciones y tiempos de respuesta [4], se pueden asignar los buses en función a su funcionalidad, quedando así en el nivel de célula los buses con mayor funcionalidad y en el nivel de proceso los buses de mayor velocidad y menor funcionalidad. La clasificación intermedia de los buses oscila entre estos niveles de comunicación en función de las características que presenten.

Para ejemplificar estos niveles se ha utilizado un bus de campo UNITEL-WAY de *Telemecanique* [1], que ha permitido realizar una aproximación a la clasificación de buses anterior.

3.1.1 Buses de bajas prestaciones

Para llevar acabo esta simulación se ha realizado una implementación del proceso de tratamiento de piezas utilizando tareas periódicas, es decir cada cierto tiempo el maestro pregunta a los esclavos por sus entradas y salidas, simulando de esta manera un bus de proceso.

Centrado en el ejemplo propuesto se ha realizado la implementación que puede observarse en la figura 6, donde el maestro controla la grúa y los esclavos controlan respectivamente los sensores superiores e inferiores del sistema donde la interacción con el sistema se realiza a partir de dispositivos como los mostrados en la figura 5.

3.1.2 Buses de altas prestaciones

En este nivel los dispositivos a utilizar tienen una mayor funcionalidad. Para realizar la implementación se ha modificado el ejemplo de partida como se puede ver en la figura 7.

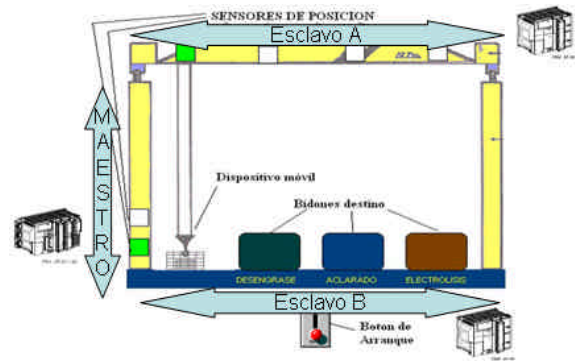


Figura 6: Pantalla de visualización limpieza electrolítica

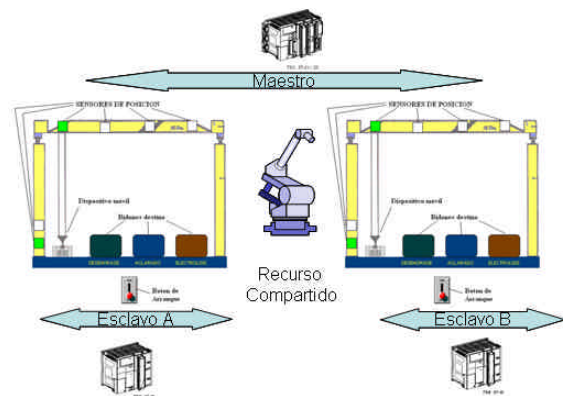


Figura 7. Esquema nivel de célula

Se ha implementado una estación de tratamiento en cada autómatas esclavo, quedando el maestro para la gestión del recurso compartido que se encarga de cargar y descargar las grúas de limpieza. Además se supone que la cuba electrolítica (la última), debe estar a una determinada temperatura, por lo que en los autómatas esclavos, se ha utilizado un módulo PID para realizar el control de la temperatura, ejemplificando de esta manera una mayor funcionalidad del sistema.

3.2 NIVEL DE INFORMÁTICA

En esta área se encuentran los niveles de planta y factoría donde la interacción con los dispositivos se realiza mediante interfaces software estandarizados. En el nivel de planta son utilizadas diversas interfaces software que pueden ser clasificadas en: Software propietario, Software empujado [21], OPC [3].

Esta clasificación ha sido ubicada en el nivel de planta por su interacción directa con el dispositivo y por ofrecer mejores prestaciones en cuanto a tiempos de respuesta.

El software propietario es el que proporciona el fabricante de los dispositivos para la interacción con el dispositivo.

En el ejemplo concreto el software propietario es el programa PL7 de *Schneider-Telemecanique* [18] que permite configurar e implementar el bus de campo y diseñar pantallas de explotación como la mostrada en la figura 6. Además también provee una librería de funciones que permite la interacción con los dispositivos (Autómatas del fabricante) desde cualquier lenguaje de programación bajo la plataforma *Microsoft Windows*, lo que permite integrar la comunicación directa con el dispositivo desde cualquier aplicación. Se ha realizado una implementación utilizando las herramientas de *LabVIEW* [19] para desarrollar una aplicación que utilizando la librería de funciones, interactúa con el dispositivo. Puede verse la implementación realizada en *LabVIEW* en la figura 8.



Figura 8. Implementación en *LabVIEW* utilizando librería de funciones

Continuando con la clasificación del software de interacción se encuentran dispositivos que utilizan software empotrado para proveer servicios de configuración e interacción como servidores Web, plataforma *JAVA* etc. Para ilustrar este tipo de dispositivos se ha utilizado un Autómata Premium que posee un servidor Web, y una implementación *JAVA* que pueden ser modificados mediante el software *Factory Cast* [20] (que acompaña al dispositivo). Estos servicios pueden analizarse en la figura 9.

Como último punto a destacar en la clasificación realizada se encuentra OPC (Ole for Process Control) que es un estándar de comunicación en el campo del control y la supervisión de procesos. Permite que diversas fuentes de datos envíen datos a un mismo servidor. La mayoría de los fabricantes del mercado disponen de un servidor OPC para interactuar con sus dispositivos.

Para poner de manifiesto la utilización de esta tecnología y continuando con el desarrollo del ejemplo elegido, se ha utilizado el software *OFS* [22] (Object Factory Server), OPC del fabricante *Schneider-Telemecanique*, que permite la interacción con los autómatas que intervienen el proceso.

Para la mencionada implementación se ha utilizado una herramienta de configuración que pertenece al conjunto de herramientas de que esta provisto *OFS* (Figura 10).

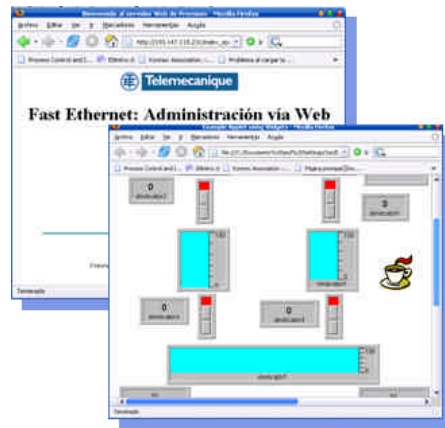


Figura 9. Software empotrado

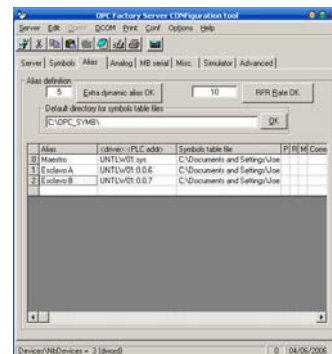


Figura 10. OFSConf

A partir de esta herramienta se configura el acceso a las aplicaciones que se encuentran ejecutando en los dispositivos.

A continuación se han utilizado una librería de funciones para implementar clientes OPC. Así se han realizado dos implementaciones. La primera se ha realizado utilizando el conjunto de herramientas de *LabVIEW* para conexión OPC (Figura 11), y la segunda utilizando *Java* (Figura 12) y la librería de funciones, (integrada con *Java* mediante el *Java Native Interface*) debido a sus características de lenguaje orientado a transacciones sobre la red.

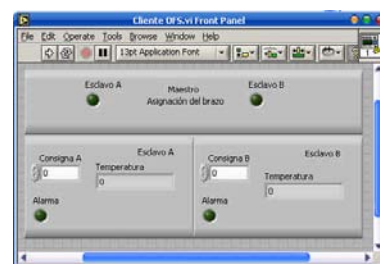


Figura 11. Implementación cliente OPC *LabVIEW*

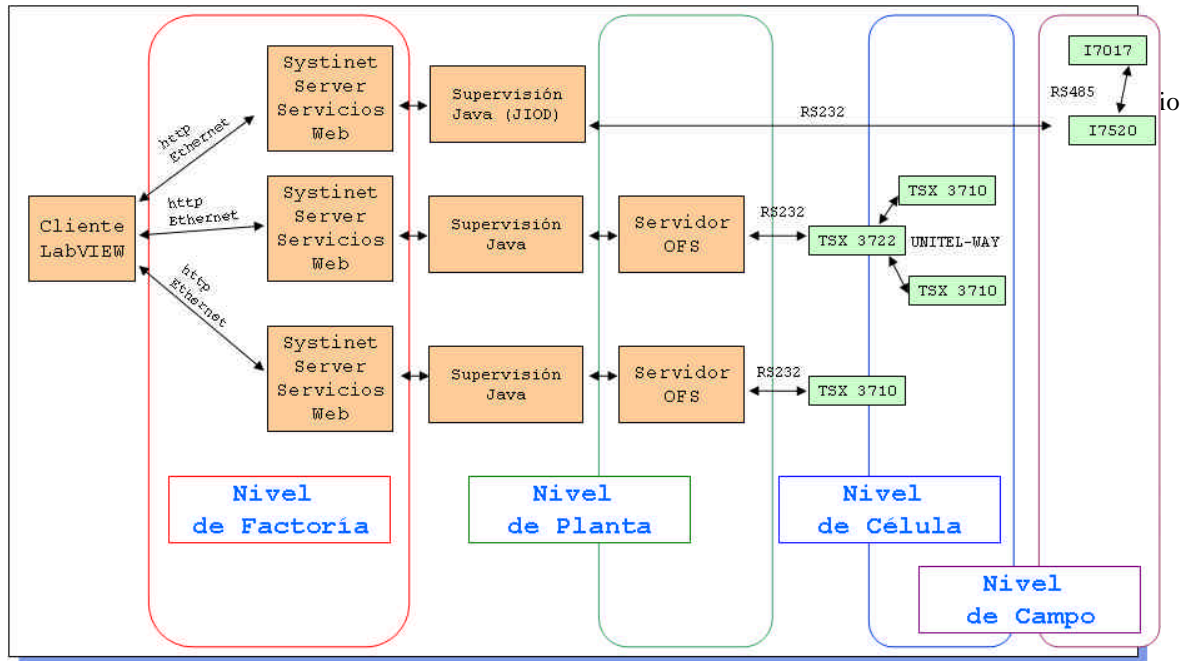


Figura 17: Esquema del sistema integrado

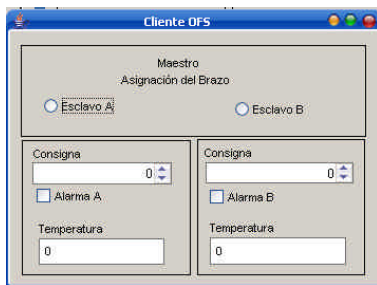


Figura 12. Implementación cliente OPC Java

En el nivel más alto de comunicación se encuentran principalmente el siguiente software de interacción:

- RMI [25]
- CORBA [25]
- Servicios Web (XML) [2] [5]

Los cuales presentan una arquitectura de comunicación como la de la figura 13:



Figura 13. Arquitectura de Comunicación a nivel de factoría

La arquitectura consiste en la implementación de un servicio que realice la interacción con los niveles

puede ser local a la aplicación o no). Desde el lado cliente, se selecciona el servicio. Una vez obtenida la referencia al servicio el cliente procede a la utilización de este.

Utilizando esta arquitectura las aplicaciones se orientan al servicio sobre la red. La aplicación ofrece sus servicios de monitorización y control a otra computadora que puede estar ubicada geográficamente en otro lugar.

Para el desarrollo del ejemplo en este nivel se ha utilizado la implementación de *OFS* realizada sobre Java, desarrollando clientes RMI, y CORBA, como puede verse en la figura 14, donde se generan archivos de configuración y de funciones compartidas tanto del lado del servidor y del cliente. Debiendo utilizar los archivos del lado cliente para desarrollar el cliente de estas tecnologías.

Los Servicios Web permiten capturar el servicio mediante herramientas locales, para su posterior utilización. Esta última opción ha sido seleccionada para su implementación en *LabVIEW*, utilizando herramientas *.NET* [23] para capturar los servicios. Puede verse un esquema general y la aplicación desarrollada en la figura 15

Por otro lado para publicar los servicios tanto RMI como CORBA poseen sus propios servidores. Para la implementación de los Servicios Web se ha utilizado *Systinet Server for Java* [24].

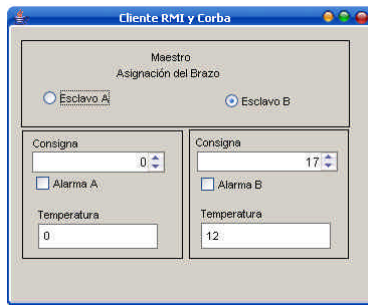


Figura 14. Implementación cliente RMI y CORBA

3.4 INTEGRACIÓN

Como se ha podido observar en punto anterior se ha implementado la integración de todos los niveles de la pirámide de automatización. En la figura 15 se puede ver el esquema de este nivel de integración así como la imagen de la aplicación desarrollada. El siguiente nivel de integración es la cooperación de sistemas sobre la red.

Una vez definido el planteamiento de la orientación al servicio los sistemas pueden cooperar y este ámbito de cooperación en este trabajo se ha definido sobre una herramienta SCADA [26]. Para completar el desarrollo de la integración se ha implementado una herramienta SCADA en LabVIEW, que permite la interacción con dispositivos distintos, para la puesta en funcionamiento se ha desarrollado una herramienta SCADA sobre LabVIEW, que se puede ver en la figura 16

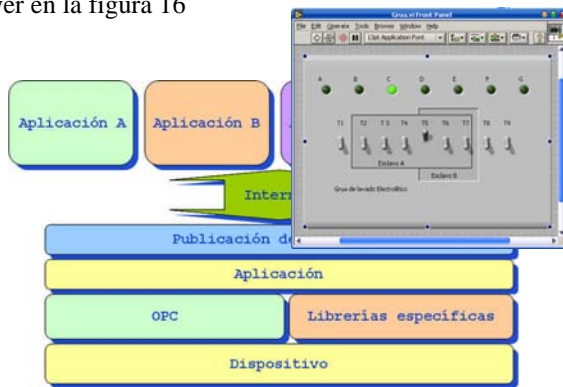


Figura 15. Implementación cliente Servicios Web en LabVIEW

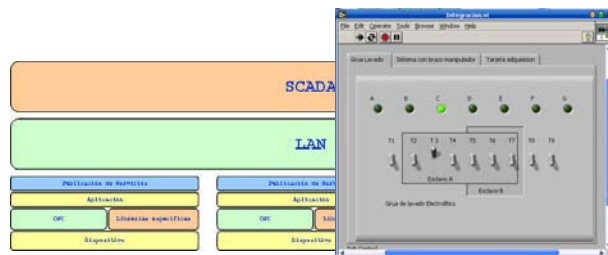


Figura 16. SCADA con integración de servicios

En la figura 17 se puede ver un esquema de la aplicación desarrollada, donde se puede observar un desglose de las tecnologías utilizadas según el nivel de comunicación.

En el nivel de campo una comunicación RS232 con las tarjetas de ICPDAS. A nivel de célula se puede ver una comunicación con el bus UNI-TELWAY. Subiendo un nivel se encuentra el OPC de comunicación con el bus de campo y con el autómatas (*OFS*). Para la comunicación con la tarjeta de E/S se ha utilizado software del fabricante. Llegado este punto se pueden supervisar los procesos mediante una aplicación en un PC de forma local, lo que permite la interacción con bases de datos para guardar históricos, utilizar software para hacer cálculos más complejos etc. Para alcanzar el nivel de factoría se ha utilizado el software *Systinet Server for JAVA* que permite publicar los servicios locales (implementados en *JAVA*), sobre TCP-IP.

De esta manera se han integrado bajo un mismo sistema redes heterogéneas.

En este punto surge un elemento de integración de servicios un servidor de polling que pregunta a cada uno de los subsistemas los servicios que están disponibles. Y surgen dos variantes:

- Productor Consumidor, el servidor ofrece los servicios a un cliente externo facilitando ubicación y características.
- Sistema Peer, que permite balancear cargas de trabajo en procesos de producción, cuando alguno de los sistemas se encuentra en estado de mantenimiento.

4 RESULTADOS

4.1 DESARROLLO DE UNA CÉLULA DE FABRICACIÓN FLEXIBLE.

Los desarrollos anteriormente descritos han sido utilizados para el desarrollo de una célula de fabricación flexible cuyo esquema se puede ver en la figura 18. (La implementación real puede verse en [8])

La célula está diseñada para llenar un estuche con artículos tipo A o tipo B (en función de los sensores) y seguidamente para proceder a su almacenamiento.

La célula consta de 4 estaciones:

- La primera espera el palet, y si está vacío pone la base del estuche
- La segunda estación comprueba el tipo de pieza y además que haya base, de ser así rellena.

- La tercera estación verifica que haya contenido y si procede pone la tapa del estuche.
- La cuarta estación verifica que la tapa esté puesta y si procede clasifica y almacena el estuche.

Para llevar a cabo esta implementación se ha utilizado un bus de campo UNITEL-WAY, un brazo manipulador SCORBASE-V [28] y un simulador de un brazo robotizado SCORBASE-V plus [28].

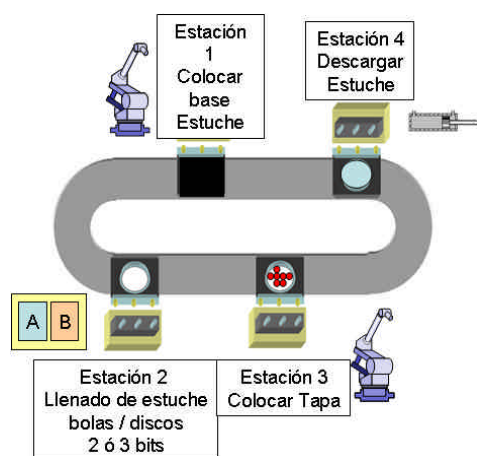


Figura 18. célula de fabricación flexible.

El bus de campo se encarga de gestionar las estaciones 2 y 4, un autómata TSX Micro 3710 [29] [8] se encarga de la interacción el brazo robotizado de la estación 1 y la estación 3 se ha utilizado un simulador. La implementación física de la interacción con el brazo robotizado puede verse en la figura 19.



Figura 19. Implementación física interacción brazo manipulador

4.2 MODELADO E IMPLEMENTACIÓN

El modelado de las estaciones se ha realizado mediante redes de Petri [27], y la implementación ha sido llevada a cabo con el software PL7. En la figura 20 se puede ver el modelado de la estación 3.

4.3 INTERACCIÓN Y SERVICIO

La interacción con los dispositivos se ha realizado a través de OPC, en concreto de *OFS* (como se indicó en la sección anterior).

Una vez configurado *OFS*, se ha procedido a implementar cada uno de los servicios de las estaciones, desarrollando clientes OPC, y creando Web Services mediante Systinet Server for Java.

4.3.1 INTEGRACIÓN DEL SISTEMA

Para la integración del sistema se ha utilizado LabVIEW para desarrollar la herramienta SCADA que supervisa el sistema. Esta herramienta puede verse en la figura 21.

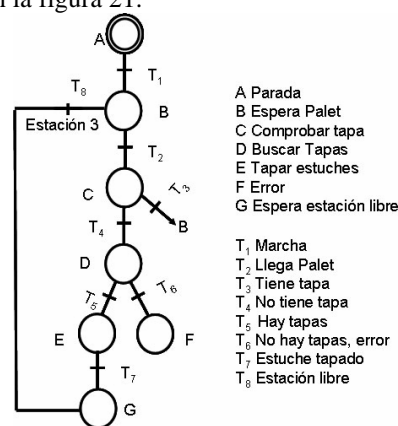


Figura 20. Ejemplo de modelado



Figura 21. Herramienta de supervisión del sistema

5 CONCLUSIONES Y TRABAJOS FUTUROS

En la actualidad y con la mejora tecnológica las redes de control tienden a estandarizarse sobre TCP/IP relegando a redes de comunicación como los buses de campo a niveles de automatización inferiores (donde los requisitos sean más estrictos).

La utilización de una arquitectura abierta en comunicaciones industriales posibilita la independencia de dispositivos. Las arquitecturas de comunicación industrial están avanzando hacia la estandarización de sistemas sobre TCP-IP [1] y hacia la orientación al servicio para aumentar la flexibilidad de sus sistemas.

Dentro del marco del trabajo desarrollado, se ha establecido varios trabajos futuros:

- Desarrollo de una herramienta de simulación de buses de campo, a la que poder imponer restricciones adecuando los buses a las necesidades de los proyectos.
- Aplicar técnicas de control a este tipo de arquitecturas abiertas, debido a los problemas sobre saturación de tráfico de las redes de datos.

Referencias

- [1] Piedrafita Moreno R., (2004) Ingeniería de la Automatización Industrial. Editorial RA-MA.
- [2] Rodríguez, D.; Sánchez, J. M.; Duran, A (2005) "Distributed Reconfigurable Computing using XML Web Services", *IEEE SIPS*, pp. 613-617.
- [3] Pérez F. (2004) OPC Conceptos Fundamentales. Dpto. de Ingeniería de Sistemas y Automática. Escuela Técnica Superior de Ingeniería. Universidad de Bilbao.
- [4] Enguita J. M. (2006) Buses de Campo. Redes Industriales. Área de Ingeniería de Sistemas y Automática. Departamento de Ingeniería Electrónica de Computadores y Sistemas. Universidad de Oviedo.
- [5] Aiello M (2006) "The Role of Web Services al Home". *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*.
- [6] Tanenbaum, A. S. (1996) Computer Networks 3rd ed., Upper Saddle River, New Jersey: Prentice Hall PTR.
- [7] Wollschlager M.; Neuman P.; Bangeman T. (2002) "Web Services for Remote Maintenance of FieldBus Based Automations Systems". *IEEE Africon*. Pp. 247-252.
- [8] Reynard S. (2005) Supervisión, control y gestión de una célula de fabricación flexible de producción mediante un sistema SCADA, Proyecto fin de Carrera, Universidad Politécnica de Cataluña.
- [9] Rafeas J.; Gomis O.; Galceran S.; Benítez I. (2005). "Autómatas Programables: estandarizar es avanzar". *Automática e Instrumentación, Informática en la Industria* Vol. 366 pp. 86-104.
- [10] Cena G.; Durante L.; Valenzano A. (1995) "Standard field bus networks for industrial applications". *Computer Standards & Interfaces*. Vol. 17 pp. 155-167.
- [11] Kim K.; Jeon G.; Hong S.; Kim T.; Kim S. (2000) "Integrating subscription-based and connection-oriented communications into the embedded CORBA for the CAN bus" *Real-Time Technology and Applications Symposium, 2000. RTAS 2000. Proceedings. Sixth IEEE*. pp. 178 - 187
- [12] Krishna, A.S.; Schmidt, D.C.; Klefstad, R.; (2004) "Enhancing Real-time CORBA via Real-time Java features". *Distributed Computing Systems, 2004. Proceedings. 24th International Conference*. pp. 66 - 73
- [13] Wen-Der Jiang; (1997) "Experience of applying CORBA middleware to air traffic control automation systems". *Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE*. Vol. 1 pp. 8-15
- [14] Li Zheng; Nakagawa, H.; (2002) "OPC (OLE for process control) specification and its developments". *SICE. Proceedings of the 41st SICE Annual Conference*. Vol. 2 pp. 917-920.
- [15] Blasco I.; (2003) Almacén Intermedio de la Célula de Fabricación Flexible, Proyecto Fin de Carrera, Universidad de Zaragoza.
- [16] Romera J.P.; Lorite J. A; Montoro S.; (2002) Automatización Problemas resueltos con autómatas programables, International Thomson Editores Spain, Paraninfo, S.A.
- [17] 7000 Series User's Manual. Extraído el 5 de abril de 2006 de <http://www.icpdas.com/download/7000/manual.htm>.
- [18] Schneider Electric. (2006) PL7 V4.3 Documentación Técnica. [Versión electrónica] Extraído el 4 de febrero de 2006 de <http://www.schneiderautomation.com>
- [19] National Instruments (1998) LabVIEW User Manual 1 [Versión Electrónica]. Extraído 15

- Mayo de 2006 de <http://proton.ucting.udg.mx/tutorial/>
- [20] Schneider Electric. (2002) Manual de Usuario de Factory Cast [Versión Electrónica]. Extraído el 4 de febrero de 2006 de <http://www.schneiderautomation.com>
 - [21] Métodos de Estimación y Evaluación del Software. Extraído el 4 de febrero de 2006 de http://html.rincondelvago.com/software_1.html
 - [22] Schneider Electric. (2002) Manual referencia de OFS v2.0 [Versión Electrónica]. Extraído el 12 de Noviembre de 2005 de <http://www.schneiderautomation.com>
 - [23] Cornwell R.; 2004 LabVIEW and .NET. Using Microsoft .NET in LabVIEW VIs. Extraído el 17 de Marzo de 2006 de <http://www.ni.com>
 - [24] Systinet; (2004). Systinet Server for Java Primer. Extraído el 17 de Marzo de 2006 de <http://www.systinet.com>
 - [25] Sun Microsystems; Java Tutorial. Extraído el 17 de Enero de 2004 de <http://www.programacion.com/java/tutorial/dcbook/>
 - [26] A. Daneels; (1999) What is SCADA?, *International Conference on Accelerator and Large Experimental Physics Control Systems*. pp. 339-343
 - [27] E. Jiménez. (2000) Redes de Petri de supervisión y simulación de procesos industriales automatizados. Dpto. Ingeniería Eléctrica. Área de Ingeniería de Sistemas y Automática. Universidad de Oviedo.
 - [28] ORPI S, L, SCORBASE-V. Extraído el 17 de Mayo 2006 de http://www.infonegocio.com/orpi/eshed/eshed_educacion.htm#er5p
 - [29] Schneider Electric. (2002) Autómatas TSX Micro, Manual de puesta en Marcha [Versión Electrónica]. Extraído el 3 de Noviembre de 2005 de <http://www.schneiderautomation.com>

ESTIMADOR DE LA FUERZAS Y PARES DE CONTACTO EJERCIDOS POR UN ROBOT MANIPULADOR SOBRE SU ENTORNO

Gámez García J.[†], Robertsson A.[‡], Gómez Ortega J.[†], Johansson R.[‡]

[†] Grupo Robótica, Automática y Visión por Computador. Universidad de Jaén, Jaén, España.
e-mail: {jggarcia, juango}@ujaen.es

[‡] Departamento de Control Automático, Universidad de Lund, Lund, Suecia.
e-mail: {Anders.Robertsson, Rolf.Johansson}@control.lth.se

Resumen

Normalmente, en aquellas operaciones robóticas en las que un robot manipulador interactúa con su entorno resulta de extrema importancia poder controlar las fuerzas y pares que éste realiza. Para poder llevar a cabo este control, se suele colocar un sensor de fuerza en la muñeca del manipulador que cierre el lazo de control. Sin embargo, en lo que se refiere a estos sensores, el principal problema que plantean es que sus medidas están influenciadas no sólo por las fuerzas y pares de contacto, sino también por aquellas fuerzas y pares relacionadas con la dinámica de la herramienta del manipulador.

El presente artículo propone el desarrollo de una nueva estrategia de fusión sensorial aplicada al problema de la estimación de las fuerzas y pares de contacto ejercidas por un robot manipulador a su entorno. Esta nueva técnica, fundamentada en la utilización de observadores de estado, recurre a una notación basada en tóresos la cual simplifica el planteamiento de la dinámica del sistema, y, consiguientemente, reduce la complejidad del estimador.

Palabras clave: Control de Fuerza y Par, Fusión Sensorial, Robots Manipuladores.

1. INTRODUCCIÓN

Es bien conocido que para un robot manipulador sin sensores en su extremo final, la posición de éste ha de seguir una trayectoria sin otra realimentación que la que ofrecen sus sensores de posición de las articulaciones, implicando importantes restricciones en la ejecución de tareas donde sea necesaria la interacción entre el robot y su entorno. Sin embargo, con la ayuda de sensores, un sistema manipulador exhibe cierta capacidad adaptativa [5], esto es, el robot se amolda a las características de su entorno, posibilitando la realización de tareas más complejas.

Por otro lado, la utilización únicamente de sensores

de fuerza/par para controlar la interacción entre el robot y su entorno proporciona una solución inicial que adolece de ciertos problemas. Probablemente, uno de los más importantes sea la aparición de perturbaciones producidas por las fuerzas y pares de inercia, efecto que se pone especialmente de manifiesto en aplicaciones con herramientas pesadas. De esta forma, la utilización de estimadores de la fuerza/par de contacto basados en observadores que fusionan la información de las medidas de fuerza/par, generadas por un sensor convencional, y de las medidas de aceleración, proporcionadas por un acelerómetro, suponen una sustancial mejora a la hora de estimar esta conjunto de fuerzas y pares [7].

En la literatura actual podemos encontrar diferentes soluciones a este problema. Inicialmente, Uchiyama et al. [11] propusieron la utilización del error de la trayectoria como datos de entrada para la estimación de las fuerzas y pares ejercidos por el manipulador a su entorno. Más tarde, el mismo Uchiyama [10], al igual que Lin en [9], desarrollaron un estimador de la fuerza de contacto basado en un filtro de Kalman Extendido. El principal inconveniente de estas soluciones, a parte de demostrarse sólo teóricamente, es que parte de sus variables, como es el caso de las aceleraciones de la herramienta, eran estimadas a partir de las medidas de posición de las articulaciones y del modelo cinemático del brazo robot.

En [1] se propuso un observador de la fuerza de contacto en la que se fusionaban las medidas de fuerza y de posición, junto con las de aceleración tomadas a partir de un acelerómetro, para estimar la fuerza normal a la superficie de trabajo. Este observador fue aplicado exitosamente en un robot manipulador con arquitectura abierta con un lazo de control de control de impedancia. Posteriormente, en [2], se extendió el observador incluyendo un acelerómetro de tres grados de libertad; por un lado el estimador trabajaba en los tres grados de libertad cartesianos; y por otro, se incluyó el modelo lineal del brazo robot dentro del observador. También en este caso se demostró su funcionamiento implementándolo en un lazo de control de impedancia

para un robot con arquitectura abierta.

La principal contribución de este trabajo es la propuesta teórica de una aplicación de fusión sensorial consistente en la integración de las medidas de fuerza/par, aceleración, posición y orientación, con el objetivo de estimar de forma apropiada las fuerzas y pares externos ejercidos por el manipulador y mejorar por tanto la ejecución de la tarea en la que el robot interacciona con el entorno. Las medidas de las que parten el observador propuesto son: medidas de fuerza/par, tomadas por un sensor colocado en la muñeca del robot; medidas de aceleración lineal y angular, tomadas por un acelerómetro lineal y un giroscopo colocados sobre la herramienta del robot; y por último, las de posición y orientación de la herramienta, que son obtenidas a partir del modelo cinemático del manipulador y de las medidas de los sensores de posición de las articulaciones.

Finalmente, el estimador desarrollado consta de un observador basado en el Filtro de Kalman en el que la dinámica de la herramienta ha sido definida utilizando torsores con el fin de facilitar el manejo de ésta.

La formulación del problema es como sigue: cuando un robot manipulador se desplaza, bien por el espacio libre, bien por el espacio restringido, un sensor de fuerza/par colocado entre la herramienta y el extremo final del manipulador no sólo medirá las fuerzas y pares ejercidas por el robot manipulador a su entorno, sino que también sus medidas vendrán afectadas por las fuerzas y pares debidas a la dinámica de la herramienta, esto es, las fuerzas y pares producidos por la inercia de la herramienta. De esta forma, la medidas de fuerza y par tomadas por el sensor (\vec{F}_s y \vec{N}_s) serán igual a las fuerzas y pares externos (\vec{F}_E y \vec{N}_E), más las fuerzas y pares debido a la inercia de la herramienta (\vec{F}_I y \vec{N}_I) (Ec. 1).

$$\begin{pmatrix} \vec{F}_s \\ \vec{N}_s \end{pmatrix} = \begin{pmatrix} \vec{F}_c \\ \vec{N}_c \end{pmatrix} + \begin{pmatrix} \vec{F}_I \\ \vec{N}_I \end{pmatrix} \quad (1)$$

Normalmente, la realización de la tarea requiere el control de las fuerzas y pares de contacto o externas (\vec{F}_c y \vec{N}_c).

Este trabajo se estructura de la siguiente forma. En el apartado II se presenta el modelo dinámico de la herramienta, describiendo los sistemas de referencia utilizados. En la sección III se describe el observador propuesto, así como la solución utilizada para estimar las fuerzas y pares de contacto. El apartado IV muestra una breve discusión, mientras que la sección V recoge las conclusiones de este trabajo.

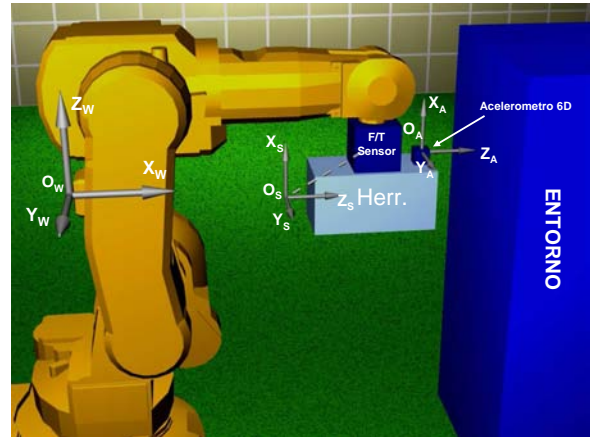


Figura 1: Sistemas de referencia definidos.

2. MODELADO DINÁMICO DE LA HERRAMIENTA

2.1. TRANSFORMACIÓN FUERZA-PAR

El conjunto de fuerzas F ($F \in \mathbb{R}^{3 \times 3}$) y pares N ($N \in \mathbb{R}^{3 \times 3}$) que ocurren en el sistema A y que está referenciados a ese sistema, puede ser definido a partir de un vector compuesto por el conjunto de fuerzas y pares como (en terminología inglesa: *wrench vector*):

$${}^A U_A = \begin{pmatrix} {}^A F_A \\ {}^A N_A \end{pmatrix} \quad (2)$$

A partir de este vector, podemos representar estas fuerzas y pares en un nuevo sistema de referencia B aplicando la siguiente transformación conocida como *wrench transformation* [6]:

$$\begin{pmatrix} {}^B F_A \\ {}^B N_A \end{pmatrix} = \begin{pmatrix} {}^B R_A^T & 0_{3 \times 3} \\ -{}^B R_A^T p^x & {}^B R_A^T \end{pmatrix} \begin{pmatrix} {}^A F_A \\ {}^A N_A \end{pmatrix} \quad (3)$$

donde ${}^B R_A$ es la matriz de rotación que relaciona el sistema A con respecto al sistema B y el vector ${}^B p^x$ se obtiene a partir de [6]:

$$p \times q = p^x q, \quad p^x = \begin{pmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{pmatrix} \quad (4)$$

2.2. SISTEMAS DE REFERENCIA UTILIZADOS

Tal y como se muestra en la figura 1, considérese un robot manipulador al que se le ha colocado un sensor de fuerza/par en la muñeca además de un acelerómetro de seis grados de libertad en su herramienta. En este caso, definimos los siguientes sistemas de referencia: $O_s X_s Y_s Z_s$ equivale al sistema de coordenadas del sensor de fuerza/par; $O_A X_A Y_A Z_A$ representa al sistema de

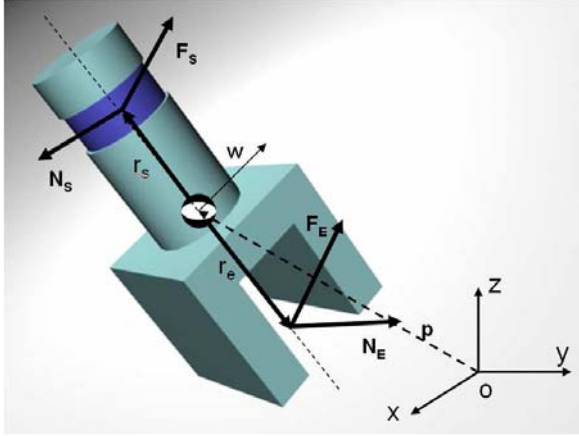


Figura 2: Fuerzas y momentos aplicados a la herramienta.

coordenadas del acelerómetro de seis grados de libertad; $O_E X_E Y_E Z_E$ equivale al sistema de coordenadas donde se realizan las fuerzas y pares externos; y finalmente, $O_W X_W Y_W Z_W$ representa el sistema de referencia global.

Por otro lado, considerando de forma local la herramienta, las fuerzas y pares externos que aparecen ejercidos sobre ella se muestran en la figura (Fig. 2), donde $O_P X_P Y_P Z_P$ representa el centro de gravedad de la herramienta.

2.3. DINÁMICA DE LA HERRAMIENTA

Como se dijo anteriormente, si colocamos un sensor de fuerza/par entre la herramienta del robot y la muñeca de ésta, este sensor no sólo medirá las fuerza y pares externos producidos por la interacción con el entorno, si no que también reflejará las perturbaciones producidas por la dinámica de ésta. Esto es, la propia dinámica de la herramienta produce en el sensor medidas de fuerza y par diferentes de las contacto o externas.

Consideremos la dinámica de la herramienta a partir de la figura 2. Además de los sistemas de coordenadas descritos en el apartado anterior, definiremos $O_P X_P Y_P Z_P$ como el sistema de referencia del centro de gravedad de la herramienta.

Por un lado, el sensor de fuerza/par medirá las fuerzas y pares (${}^S F_S$ y ${}^S N_S$) que se producen entre el brazo del manipulador y la herramienta. Por otro lado, las fuerzas y pares ejercidas por el entorno sobre la herramienta vienen definidas por (${}^E F_E$ y ${}^E N_E$). Entonces se deduce que las fuerzas y pares ejercidas sobre el centro de gravedad vendrían dadas a partir de las si-

guientes ecuaciones:

$$\begin{aligned} {}^P F_P &= {}^P R_S \cdot {}^S F_S + {}^P R_E \cdot {}^S F_E \\ {}^P N_P &= {}^P R_S \cdot {}^S N_S + {}^P r_S \times {}^P R_S \cdot {}^S F_S \\ &\quad + {}^P R_E \cdot {}^E N_E + {}^P r_E \times {}^P R_E \cdot {}^E F_E \end{aligned} \quad (5)$$

Utilizando la notación propuesta en la ecuación (2), la Ec. (5) queda convertida a:

$$\begin{aligned} {}^P W_P &= \begin{pmatrix} {}^P F_P \\ {}^P N_P \end{pmatrix} = \begin{pmatrix} {}^P R_S^T & 0_{3 \times 3} \\ -{}^P R_S^T p^x & {}^P R_S^T \end{pmatrix} \begin{pmatrix} {}^S F_S \\ {}^S N_S \end{pmatrix} \\ &\quad + \begin{pmatrix} {}^P R_E^T & 0_{3 \times 3} \\ -{}^P R_E^T p^x & {}^P R_E^T \end{pmatrix} \begin{pmatrix} {}^E F_E \\ {}^E N_E \end{pmatrix} \\ &= {}^P T_S \begin{pmatrix} {}^S F_S \\ {}^S N_S \end{pmatrix} + {}^P T_E \begin{pmatrix} {}^E F_E \\ {}^E N_E \end{pmatrix} \end{aligned} \quad (6)$$

donde ${}^P R_S$, ${}^P R_E$ son las matrices de rotación que relaciona los sistemas S y E con P , respectivamente. ${}^P p^x$ y ${}^P p^x$ son matrices de dimensión 3×3 y se obtienen la Ec. (4) donde el vector p representa la distancia entre el origen del sistema de referencia del centro de gravedad y el origen del sistema de coordenadas, bien del sensor de fuerza, bien del punto de aplicación de la fuerza. Finalmente, aplicando las ecuaciones de Newton-Euler, la dinámica de la herramienta queda definida como

$$\begin{aligned} {}^P W_P &= \begin{pmatrix} m^w R_P^{-1} (a - g_o) \\ I \dot{\omega} + \omega \times I \omega \end{pmatrix} \\ &= \begin{pmatrix} m^w R_P^{-1} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{pmatrix} \begin{pmatrix} a \\ \dot{\omega} \end{pmatrix} \\ &\quad + \begin{pmatrix} m^w R_P^{-1} & 0_{3 \times 3} \\ 0_{3 \times 3} & \omega \times I \end{pmatrix} \begin{pmatrix} g_o \\ \omega \end{pmatrix} \end{aligned} \quad (7)$$

donde m es la masa total de la herramienta; $a = [\ddot{x}, \ddot{y}, \ddot{z}]^T$ es la aceleración de la herramienta referida al sistema global O_W ; $g_o = [g_x, g_y, g_z]^T$ es la aceleración de la gravedad; $\omega = [\omega_1, \omega_2, \omega_3]^T$ es la velocidad angular de la herramienta en torno a los ejes x_p, y_p y z_p y la matriz I es igual a

$$I = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad (8)$$

donde I_1, I_2 e I_3 son los principales momentos de inercia con respecto a x, y, z .

3. ESTIMADOR DE LAS FUERZAS Y PARES DE CONTACTO

El objetivo principal de este trabajo es el de proponer, de forma teórica, un observador de la fuerza y par de contacto ejercido por un robot manipulador a su entorno. A diferencia de trabajos anteriores [10][9][8], en este artículo se propone que una de las variables medidas, además de las ya habituales: posición, orientación, fuerza y par; sean la aceleración lineal y angular de la herramienta.

El primer paso para obtener un estimador de las fuerzas y pares externos será el de definir la dinámica de la herramienta a través de espacio de estados. Para ello, definimos a continuación el vector de estado como:

$$X = [x, y, z, \theta, \phi, \psi, \dot{x}, \dot{y}, \dot{z}, w_1, w_2, w_3] \quad (9)$$

donde x, y, z son las coordenadas de posición del centro de gravedad de la herramienta y θ, ϕ y ψ son los ángulos de Euler que define la orientación de la herramienta. Ambos están referidos a $O_W X_W Y_W Z_W$. Por otro lado, $\dot{x}, \dot{y}, \dot{z}$ corresponden con las velocidades de la posición del centro de gravedad y en cuanto a los ángulos de Euler considerados, estos se relacionan con las velocidades angulares $\omega = [\omega_1, \omega_2, \omega_3]^T$ de la herramienta a través de la siguiente ecuación [3]:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} = A_1 \cdot \omega = \begin{pmatrix} \frac{\sin(\psi)}{\cos(\theta)} & \frac{\cos(\psi)}{\sin(\theta)} & 0 \\ \frac{\cos(\psi)}{\tan(\theta)} & -\frac{\sin(\psi)}{\tan(\theta)} & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \quad (10)$$

La ecuación en espacio de estados que define nuestro sistema sería:

$$\dot{X} = A(X)X + g + B_S U_S + B_E U_E \quad (11)$$

donde $U_S = [{}^S F_S, {}^S N_S]^T$, $U_E = [{}^E F_E, {}^E N_E]^T$, $g_h = [0_{3 \times 1}, g_o, 0_{6 \times 1}]^T$ y las matrices $A(X)$, B_S y B_E obtienen sus valores a partir de las ecuaciones (6) y (7), resultando:

$$A(X) = \begin{pmatrix} 1 & 0 & 0 & 0_{3 \times 9} \\ 0 & 1 & 0 & A_{1 \times 3} \\ 0 & 0 & 1 & A_{2 \times 3} \\ 0_{9 \times 9} & A_{1 \times 3} & A_{2 \times 3} & A_{3 \times 3} \end{pmatrix}; \quad (12)$$

$$B_S = \begin{pmatrix} 1/m^w R_P & 0_{3 \times 3} \\ 0_{3 \times 3} & I^{-1} \end{pmatrix} \cdot {}^P T_S;$$

$$B_E = \begin{pmatrix} 1/m^w R_P & 0_{3 \times 3} \\ 0_{3 \times 3} & I^{-1} \end{pmatrix} \cdot {}^P T_E$$

donde la matriz A_2 es igual a:

$$A_2 = \begin{pmatrix} 0 & w_3 \frac{I_2 - I_3}{I_1} & 0 \\ 0 & 0 & w_1 \frac{I_3 - I_1}{I_2} \\ w_2 \frac{I_1 - I_2}{I_3} & 0 & 0 \end{pmatrix} \quad (13)$$

Por otro lado, puesto que la posición y orientación de la herramienta, las medidas de fuerza y par del sensor

colocado en la muñeca del robot, y las aceleraciones lineales y angulares de la herramienta capturadas por los acelerómetros y giróscopos colocados en la herramienta, son las medidas con las que contamos, las proponemos como salidas en nuestro sistema. El vector de salida quedaría por tanto como:

$$y = \begin{pmatrix} p^T \\ o^T \\ F_s^T \\ N_s^T \\ \ddot{p}^T \\ \dot{w}^T \end{pmatrix} \quad (14)$$

que, utilizando las ecuaciones (6) y (7) podría convertirse a:

$$y = C(X)X + D_S U_S + D_E U_E + D_G g_o \quad (15)$$

donde las matrices $C(X)$, D_S y D_E serían igual a:

$$C(X) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0_{9 \times 6} & 0_{9 \times 6} \\ 0_{3 \times 9} & A_{2 \times 6} \\ 0_{3 \times 9} & 0_{3 \times 9} \end{pmatrix} \quad (16)$$

$$D_S = \begin{pmatrix} 0_{6 \times 6} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ D_{S1} & 0_{3 \times 6} \end{pmatrix}$$

$$D_E = \begin{pmatrix} 0_{6 \times 6} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ D_{E1} & 0_{3 \times 6} \end{pmatrix}$$

$$D_G = \begin{pmatrix} 0_{18 \times 3} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

donde

$$\begin{aligned} D_{S_1} &= \begin{pmatrix} 1/m^w R_P & 0_{3 \times 3} \\ 0_{3 \times 3} & I^{-1} \end{pmatrix} \cdot {}^P T_S; \\ D_{E_1} &= \begin{pmatrix} 1/m^w R_P & 0_{3 \times 3} \\ 0_{3 \times 3} & I^{-1} \end{pmatrix} \cdot {}^P T_E \end{aligned} \quad (17)$$

obteniéndose A_2 a partir de la ecuación (13).

Finalmente, utilizando las ecuaciones (11) y (15), el sistema en espacio de estado que representa la dinámica de la herramienta del manipulador quedaría como:

$$\begin{cases} \dot{X} = A(X)X + g + B_S U_S + B_E U_E + v_X \\ y = C(X)X + D_S U_S + D_E U_E + g_o + v_y \end{cases} \quad (18)$$

donde las matrices $A(X)$, B_S , B_E , $C(X)$, D_S y D_E se obtienen de las ecuaciones (12) y (16); g_o correspondería con el vector aceleración; v_X y v_y son, respectivamente, el ruido del proceso y de las variables de medida.

Una vez definida la dinámica, y para obtener un estimador apropiado de las fuerzas y pares externos, se propone un observador donde la entrada U_E no sea considerada [1][2]. Entonces, la diferencia entre el sistema real y el observador propuesto, dará lugar a una estimación de las fuerzas y pares de contacto. Nuestro observador sería por tanto:

$$\dot{\hat{X}} = A(X)\hat{X} + B_S U_S + K(y - \hat{y}) \quad (19)$$

$$\hat{y} = C(X)\hat{X} + D_S U_S + g_o \quad (20)$$

donde \hat{X} corresponde a la estimación de X y K es la ganancia del observador, siendo:

$$K = \begin{pmatrix} k_{1,1} & k_{1,2} & \dots & k_{1,12} \\ \dots & & & \\ k_{21,1} & k_{21,2} & \dots & k_{21,12} \end{pmatrix} \quad (21)$$

La dinámica del error de la estimación vendría dada a partir de $\tilde{X} = X - \hat{X}$ obteniéndose

$$\dot{\tilde{X}} = (A(X) - KC(x))\tilde{X} - (B_E + KD_E)U_E \quad (22)$$

$$\tilde{y} = y - \hat{y} = C(X)\tilde{X} + D_E U_E \quad (23)$$

Finalmente, si aseguramos que la matriz $(A(X) - KC(x))$ tiene sus autovalores con parte real negativa, de forma que para un valor de estado X el observador fuese estable, un estimador de las fuerzas y pares de contacto vendría dado como:

$$\hat{U}_E = D_E^\dagger (-C(X)\tilde{X} + \tilde{y}) \quad (24)$$

donde D_U^\dagger es la pseudoinversa de D_U . En este caso se cumple la propiedad

$$\begin{aligned} \hat{U}_E &= D_U^\dagger (-C(X)\tilde{X} + \tilde{y}) \\ &= D_U^\dagger (-C(X)\tilde{X} + C(X)\tilde{X} + D_E U_E) \\ &= D_E^\dagger D_E U_E = U_E \end{aligned} \quad (25)$$

implicando que $\hat{U}_E = U_E$.

3.1. FILTRO DE KALMAN EXTENDIDO

Como se puede comprobar en las ecuaciones (19) y (20) a través de las matrices $A(X)$ y $C(X)$, el problema de extraer las fuerzas y pares externos a partir de las medidas de fuerza/par del sensor, de las posiciones y orientaciones de la herramienta, y de las aceleraciones de ésta, representa un problema de estimación de estados no lineales, estimación cuya solución exacta resulta imposible de obtener [11].

Consideremos entonces el problema de la estimación de estados para una trayectoria dada \bar{X} . En este caso, las ecuaciones (19) y (20) pueden ser linealizadas utilizando las series de Taylor para una trayectoria predefinida \bar{X} , ya que esta trayectoria representaría una solución particular del sistema estocástico descrito [4]. Aplicando esta linealización tendríamos:

$$F(t) = A(\bar{X}(t)) + \frac{\partial A}{\partial X}(\bar{X}(t))\bar{X}(t) \quad (26)$$

$$H(t) = C(\bar{X}(t)) + \frac{\partial C}{\partial X}(\bar{X}(t))\bar{X}(t) \quad (27)$$

en la que estas expresiones podrían tener en cuenta, además de los términos lineales, los de mayor grado, pero debido a altas exigencias computacionales consideramos esta opción como la adecuada.

Finalmente, utilizando la información de la ecuación (18), la ganancia $K(t)$ puede ser estimada utilizando el Filtro de Kalman a partir de la siguiente ecuación:

$$K(t) = P(t)H^T(t)R^{-1}(t) \quad (28)$$

de la que $P(t)$ se obtiene como solución a la siguiente ecuación de Riccati:

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) \quad (29)$$

$$- P(t)H^T(t)R^{-1}H(t)P(t) + BQB^T \quad (30)$$

donde

$$E[v_\xi(t) v_\xi^T(\tau)] = Q(t)\delta(t - \tau); E[v_\xi(t)] = 0 \quad (31)$$

$$E[v_y(t) v_y^T(\tau)] = R(t)\delta(t - \tau); E[v_y(t)] = 0$$

$$E[v_\xi(t) v_y^T(\tau)] = S(t)\delta(t - \tau) = 0; \forall t, \tau$$

3.2. PROPIEDADES DEL ESTIMADOR PROPUESTO

Para analizar las propiedades del observador propuesto, consideremos en primer lugar la ecuación (19) transformándola a:

$$\dot{\hat{X}} = \frac{-(B_E + KD_E)}{sI - A(X) + KC(X)} \cdot U_E \quad (32)$$

Entonces, a partir del estimador definido en la ecuación (24), y utilizando la Ec. (32), tendríamos

$$\hat{U}_E = -D_E^\dagger C(X) \frac{-(B_E + KD_E)}{sI - A(X) + KC(X)} U_E + D_E^\dagger \tilde{y} \quad (33)$$

Por otro lado, desarrollando \tilde{y} teniendo en cuenta el ruido del sistema, obtendríamos:

$$\begin{aligned}\tilde{y} &= y - \hat{y} = C(X) + v_X - \hat{X} + D_S U_S + D_E U_E \\ &+ g_o + v_y - C(X) \hat{X} - D_S U_S \\ &= C(X) \tilde{X} + C(X) v_X + D_E U_E + g_o + v_y\end{aligned}\quad (34)$$

donde, considerando la ecuación (33) y sustituyendo \tilde{y} por la ecuación (34), tendríamos:

$$\hat{U}_E = D_E^+ C(X) v_X + U_E + D_E^+ v_y\quad (35)$$

Comprobándose que, aunque el estimador propuesto es insesgado, éste es sensible tanto al ruido del proceso como al introducido por las medidas, por lo que conviene diseñar la ganancia K de forma que el observador final tenga propiedades de filtro paso-bajos. Este problema queda resuelto utilizando el filtro de Kalman extendido.

4. DISCUSIÓN

El trabajo propuesto aborda el tema de robots manipuladores en interacción con su entorno. Más específicamente, se expone la problemática de la estimación de las fuerzas y pares de contacto de la herramienta terminal con una superficie de trabajo. Hasta ahora, en la mayoría de los trabajos sobre el tema de robots con interacción, las fuerzas y pares debido a las propiedades dinámicas de la herramienta son usualmente despreciadas; sin embargo, es de esperar que las perturbaciones producidas por estas fuerza y pares deterioren el comportamiento de un robot manipulador que interacciona con su entorno; más concretamente cuando éste ha de superar una fase de transición espacio libre - espacio restringido, o bien cuando la herramienta de trabajo es una herramienta pesada.

En cuanto al estimador descrito, y debido a la no-linealidad del sistema modelado, se ha propuesto utilizar una técnica basada en el *Filtro de Kalman Extendido* consistente en la linealización del sistema para unas variables de estado predefinidas. A la hora de implementar este filtro aparecen diferentes alternativas para dar valores a estas variables: modo off-line, modo on-line, etc. La más común sería linealizar el sistema a partir de una variables de estado, que en nuestro caso corresponderían con la trayectoria del robot, a partir de las trayectorias nominales. En este punto cabe la discusión de si lo más adecuado sería utilizar esta trayectoria—posición y orientación— definida de antemano y en la que no se ha tenido en cuenta la interacción con la superficie de trabajo, o bien si lo más apropiado es utilizar como valores de las variables una estimación on-line que tenga en cuenta la ejecución de la tarea hasta el instante actual. Parece claro que la segunda opción es la correcta, ya que normalmente al linealizar sobre la trayectoria nominal, la desviación entre la trayectoria actual y la nominal aumenta con el tiempo.

5. CONCLUSIONES

Para estimar apropiadamente las fuerzas y pares de contacto en situaciones donde el robot trabaja tanto en espacio libre como en espacio restringido, se ha desarrollado de forma teórica un observador que integra, por un lado, las medidas de las fuerzas y pares captadas por un sensor de fuerza/par de 6 gdl situado en la muñeca del robot; por otro, las medidas de aceleración de la herramienta obtenidas a partir de sensores de aceleración colocados sobre la herramienta; y finalmente, las medidas de posición y orientación de la herramienta situada en el extremo del manipulador.

El principal objetivo del observador propuesto es el desarrollo de un estimador de la fuerza/par de contacto que permita diseñar algoritmos de control de fuerza donde las características dinámicas de la herramienta no perturben las medidas de fuerza/par externas. Esta proposición implica la mejora de la ejecución de aquellas tareas donde fuese necesario controlar la interacción del brazo robot con su entorno.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el proyecto DPI2004-04458 del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2004/2007.

Referencias

- [1] J. Gamez, A. Robertsson, J. Gomez, and R. Johansson. Sensor fusion of force and acceleration for robot force control. In *Int. Conf. Intelligent Robots and Systems (IROS 2004)*, pages 3009–3014, Sendai, Japan, 2004.
- [2] J. Gamez, A. Robertsson, J. Gomez-Ortega, and Rolf Johansson. Force and acceleration sensor fusion for compliant robot motion control. In *IEEE Int. Conf. on Robotics and Automation (ICRA2005)*, pages 2720–2725, Barcelona, Spain, 2005.
- [3] H. Goldstein. *Classical Mechanics*. Addison-Wesley, Massachusetts, USA, 1987.
- [4] M. S. Grewal and A. P. Andrews. *Kalman Filtering. Theory and Practice*. Prentice Hall, New Jersey, USA, 1993.
- [5] F. Harashima and Y. Dote. Sensor based robot systems. In *Proc. IEEE Int. Workshop Intelligent Motion Control*, pages 10–19, Istanbul, Turkey, August 1990.
- [6] R. Johansson. *System Modeling and Identification*. Prentice Hall, Englewood Cliffs, NJ, 1993.

- [7] R. Johansson and A. Robertsson. Robotic force control using observer-based strict positive real impedance control. In *IEEE Proc. Int. Conf. Robotics and Automation*, pages 3686–3691, Taipei, Taiwan, September 2003.
- [8] M. Kumar and D. Garg. Sensor based estimation and control of forces and moments in multiple cooperative robots. *Trans. ASME J. of Dynamic Systems, Measurement and Control*, 126(2):276–283, June 2004.
- [9] S.T. Lin. Force sensing using Kalman filtering techniques for robot compliant motion control. *J. of Intelligent and Robotics Systems*, 135:1–16, 1997.
- [10] M. Uchiyama and K. Kitagaki. Dynamic force sensing for high speed robot manipulation using Kalman filtering techniques. In *Proc. of Conf. Decision and Control*, volume 12(6), pages 2147–2152, Tampa, Florida, 1989.
- [11] M. Uchiyama, M. Yokota, and K. Hakomori. Kalman filtering the 6-axis robot wrist force sensor signal. In *Proc. Int. Conf. Advanced Robotics*, pages 153–160, Tokyo, 1985.

MODELO NEUROBORROSO DE LA PRESIÓN DE LA CÁMARA DE MEZCLA EN UNA FUNDICIÓN DE COBRE

Juan Manuel Escaño

Centro Tecnológico Altair C/ El Barbero de Sevilla, 1 41006 Sevilla. jmesca@e-altair.org

Carlos Bordons

Dpto. Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingenieros.

Camino de los Descubrimientos, s/n., 41092 Sevilla. bordons@esi.us.es

Resumen

En este artículo se presenta un modelo neuroborroso de la presión de una cámara de mezclas de gases en una fundición de cobre, dentro del circuito de tratamiento de gases de la fábrica. Se ha usado el método de análisis de componentes principales para la reducción del espacio de variables de entrada y se ha obtenido además un modelo no recurrente, usándose para ello modelos neuroborrosos jerárquicos. El modelo se ha validado con datos experimentales obtenidos en una planta real.

Palabras clave: Modelos Neuroborrosos, Análisis de componentes principales, Modelado.

1 Introducción

El proceso de obtención del cobre tiene como subproducto una corriente de gases rica en azufre. Estos gases se pueden procesar para obtener ácido sulfúrico a partir de ellos. Tras la salida de los diferentes hornos donde se funde el mineral, los gases pasan por una cámara de mezcla antes de ser enviados a la fase de lavado previa a las plantas de obtención de ácido. En todo el circuito (y por tanto también en la cámara de mezcla) la presión de los gases debe ser negativa para impedir la emisión de los mismos al ambiente. Esta presión depende de diversas variables en la línea de producción y es bastante difícil de predecir, tanto obteniendo un modelo de primeros principios como por métodos de identificación. Es por tanto útil el empleo de métodos basados en las redes de neuronas artificiales o los sistemas neuroborrosos. Para realizar el modelo se tienen datos provenientes del sistema de control distribuido instalado en la planta referidos a 57 variables tomados cada 2 segundos. Para el modelo que se presenta aquí se trabajó con datos de tres días distintos, durante los que se realizaron además, pruebas en lazo abierto de diversas variables. En un primer estudio, se hicieron diversos experimentos para observar la correlación de las variables con la presión de la cámara de mezcla.

La organización del resto del artículo es la siguiente.

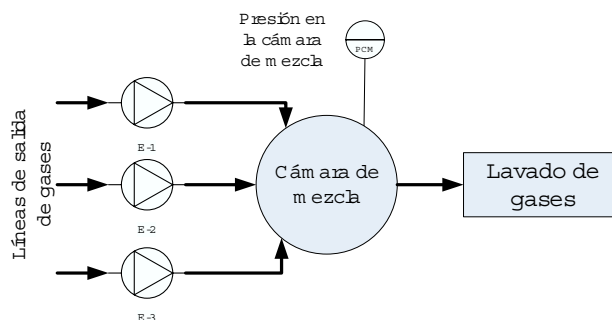


Figura 1: Mezcla de gases. Producción ácido sulfúrico

En la sección 2 se hace una breve introducción al modelado neuroborroso aplicado al sistema. En la sección 3 muestra la técnica de análisis de componentes principales, usada aquí para la reducción del espacio de variables. En las secciones 4 y 5 se muestran los resultados de los distintos modelos obtenidos. Finalmente, las conclusiones de este trabajo se resumen en la sección 6.

2 Modelado Neuroborroso

La meta del modelado borroso (Fuzzy modelling) es la obtención de un conjunto de reglas que describan la dinámica de la planta a través de unos datos experimentales. Los Sistemas de Inferencia Borrosos (FIS) son aproximadores funcionales genéricos, es decir, dado un cierto nivel de error, se puede encontrar un FIS que aproxime cualquier función con un error menor que el fijado. Para ello se pueden usar diversas técnicas, algunas provenientes del campo de las redes neuronales y otras de otros campos (estadística, algoritmos genéticos, etc). Muchos son los métodos que pueden encontrarse en la literatura para la identificación de un modelo borroso. Los métodos más comunes o conocidos son el *Fuzzy neuronal network* (FNN), el método de *Linguistic Equations* (LE) y los métodos de *Fuzzy clustering* (FC). El uso de Sistemas de Inferencia Borrosos (*Fuzzy Inference Systems*) para el modelado permite manejar bien

los procesos no lineales y dan un conocimiento del sistema que es imposible con el uso de redes neuronales. Los modelos basados en sistemas de inferencia neuroborrosos adaptativos (ANFIS) combinan la ventaja de las redes neuronales adaptativas (ANN), tales como la capacidad de aprendizaje y adaptación, y la *lógica borrosa*, es decir, el conocimiento basado en reglas y el manejo de incertidumbre y significancia del conocimiento. A diferencia de los sistemas basados en ANNs, los sistemas ANFIS pueden partir de una base de conocimiento previo, o bien, se puede añadir después la información que se desee para poder mejorar el modelo.

A la hora de establecer la estructura de un ANFIS, uno de los problemas es fijar las funciones de pertenencia. Un mayor número dará una mayor precisión en el conocimiento, pudiéndose provocar el manejo de numerosas reglas. Este problema de la explosión de reglas a mayor cantidad de términos lingüísticos, se puede subsanar con la aplicación de métodos de clustering que busquen la clasificación de datos en subconjuntos (*subset*). Además de la adecuada elección de reglas y funciones de pertenencia mediante una metodología, es importante la elección de las variables de entrada.

En un sistema simple, de pocas variables, es fácil la elección de las mismas observando la causalidad. Cuando el sistema está compuesto de muchas variables (como el caso que nos ocupa, en el que intervienen 57), acopladas entre ellas y difícilmente observable a simple vista, un método puede ser introducir el máximo número de variables en la entrada del FIS y dejar que un método de clustering decida los subconjuntos adecuados entre ellas junto con la entrada. Esto es algo, en ocasiones, poco práctico, perdiendo además una de las propiedades esenciales de los sistemas borrosos: el conocimiento del comportamiento del sistema.

En nuestro caso, queremos obtener un modelo de la variable PCM (presión de la cámara de mezcla). El camino inicial debería ser el estudio previo del sistema observando, mediante el análisis de correlaciones y los ensayos adecuados, las variables que afectan realmente a la PCM y después de ello, plantear un ANFIS con las variables mencionadas como entradas.

La metodología empleada en este trabajo es la siguiente: En primer lugar se han elegido suficientes datos para entrenar el sistema, en concreto se han usado 6000 datos de 57 variables. Se han elegido, además datos de otros días para poder validar el modelo y para evitar el sobreentrenamiento del sistema ANFIS [3]. Se han normalizado las variables, para evitar que la distinta

naturaleza y magnitud de las mismas, afecten al proceso de aprendizaje neuronal. Después se han probado algunas estructuras de sistemas ANFIS, eligiéndose las estudiadas en las secciones 4 y 5

Debido al acoplamiento dado entre las variables y la dificultad existente en la realización de pruebas experimentales, el estudio inicial no aporta la suficiente información para la implementación de un modelo adecuado. El uso del método PCA proporciona dos características: la obtención de nuevas variables incorreladas, la condensación de toda la información en un espacio mucho más reducido, proporcionando así una mayor simplificación del ANFIS. El inconveniente es la pérdida de simplicidad en el significado de las reglas del sistema borroso, pues las entradas requieren de interpretación.

3 Análisis PCA

En las situaciones donde hay un gran número de variables, es difícil obtener una visión general de las relaciones que puedan existir entre ellas. Uno de los métodos más importantes usados en el campo de la estadística multivariante es el Análisis de Componentes Principales (PCA), cuyo objetivo es reducir la dimensión del sistema, minimizando la pérdida de información.

Suponiendo p variables, la matriz de n datos es

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

Cada fila i nos dará un dato en el tiempo t_i con p variables.

Las p variables son los vectores generadores del subespacio V . Lo que perseguimos es encontrar una nueva base de vectores (Y_1, Y_2, \dots, Y_k) que definan un nuevo subespacio donde la información contenida sea la máxima. Esto es:

$$\begin{aligned} Y_1 &= w_{11}x_1 + w_{12}x_2 \dots + w_{1p}x_p \\ Y_2 &= w_{21}x_1 + w_{22}x_2 \dots + w_{2p}x_p \\ &\dots \dots \dots \\ Y_k &= w_{k1}x_1 + w_{k2}x_2 \dots + w_{kp}x_p \end{aligned}$$

Donde $k < p$.

Se demuestra que

$$\mu_Y = E(Y) = E(W^T X) = W^T E(X)$$

y que la matriz de covarianza de Y es igual a

$$\Sigma_Y = E\{(Y - \mu_Y)(Y - \mu_Y)^T\} = W^T \Sigma_X W$$

El objetivo es conseguir la máxima varianza de los datos en los ejes nuevos, es decir, la máxima Σ_Y , con la clara restricción de que sean ortonormales, esto es $W^T W = I$. En definitiva, se trata de maximizar

$$W^T \Sigma_X W - \lambda(W^T W - I)$$

derivando la expresión e igualando a cero, tenemos que

$$(\Sigma_X - \lambda I)W = 0$$

En definitiva, el problema se restringe a calcular los autovectores de Σ_X

Las componentes W_j asociadas a los autovalores con más valor de Σ_X , son las significativas para representar los datos, denominándose éstas las Componentes Principales del sistema.

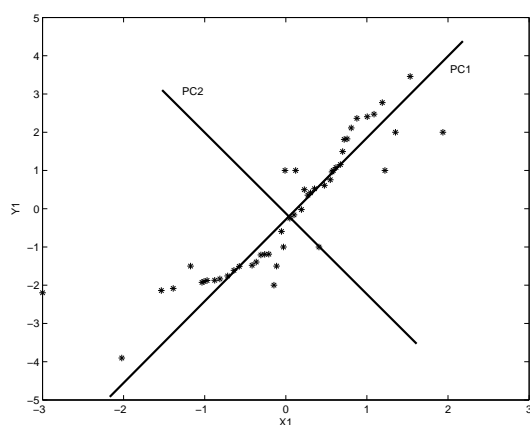


Figura 2: Componentes Principales

La primera componente es la correspondiente al mayor autovalor, la segunda al siguiente mayor, y así sucesivamente. Obteniéndose así un nuevo subespacio donde se encierra casi toda la información. A la hora de elegir las componentes principales, existen varios criterios. Uno de ellos es usar el número de autovectores l asociados a los autovectores que cumplan que

$$\frac{\sum_{i=1}^l \lambda_i}{\sum_{i=1}^n \lambda_i} \geq n$$

siendo n el porcentaje de información deseada.

Al aplicar el análisis de componentes principales al sistema en cuestión (con 57 variables), se obtuvieron 7 componentes principales que contienen el 95% de la información, reduciéndose considerablemente el espacio de las entradas al sistema

4 Modelo Dinámico

Un modelo borroso dinámico extiende el dominio de aplicación de las redes de neuronas bor-

rosas (FNN) a los problemas temporales (en el tiempo). Son llamados también Sistemas de neuronas borrosas recurrentes (RFNN). Son sistemas que poseen las mismas ventajas que las redes de neuronas recurrentes [5]. La recurrencia es debida a la realimentación de la salida para poder así captar la dinámica y el cambio. Estos sistemas presentan una dificultad: la propia estabilidad de los mismos una vez que están on-line. Los sistemas no dinámicos no tienen este problema, debido a la respuesta instantánea y la *unidireccionalidad* de la información.[6] [7]

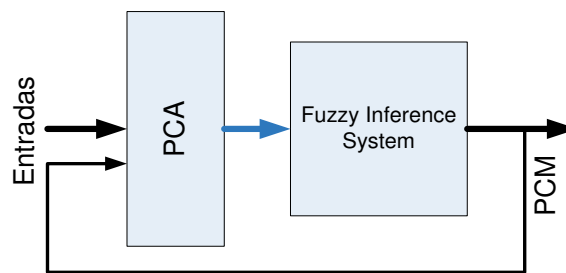


Figura 3: Modelo neuroborroso dinámico

Debido a problemas encontrados en el entrenamiento, de la red dinámica clásica propuesta, en el caso que nos ocupa, se propuso un esquema distinto -también dinámico- pero incluyendo en el análisis PCA el valor anterior de la variable de salida, realimentando indirectamente la misma. Para el ajuste se usó la toolbox de MATLAB® de lógica borrosa [3]. La tabla 1 describe las características del sistema RFNN obtenido

Tabla 1: Características del Sistema Borroso

Tipo de sistema	T-Sugeno
Nº de nodos	294
Parámetros (lineales y no lineales)	128
Número de reglas	128
Tipo de salidas	Constantes
Funciones de pertenencia	gaussianas
Funciones de pertenencia por entrada	2

En la figura 3 puede verse un diagrama de la propuesta del sistema.

En la figura 4 se presenta la validación del modelo, siendo éste representado en línea azul.

El modelo así obtenido se ha validado para distintos días de funcionamiento, en los cuales las condiciones de operación cambian continuamente. La figura 5 muestra la predicción que ofrece el modelo en otra situación.

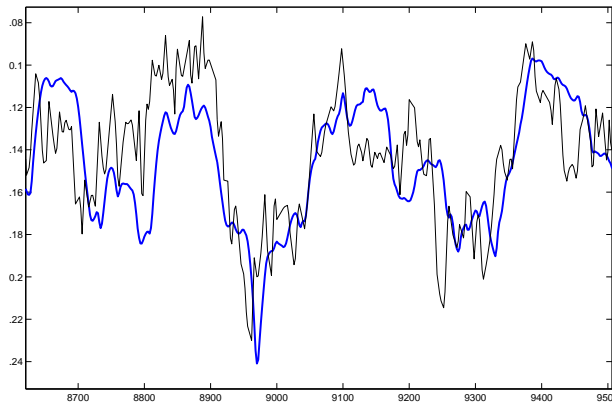


Figura 4: Resultados del modelo dinámico

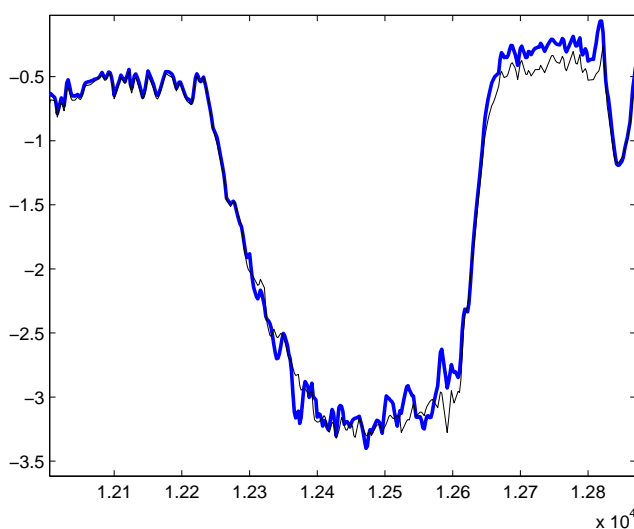
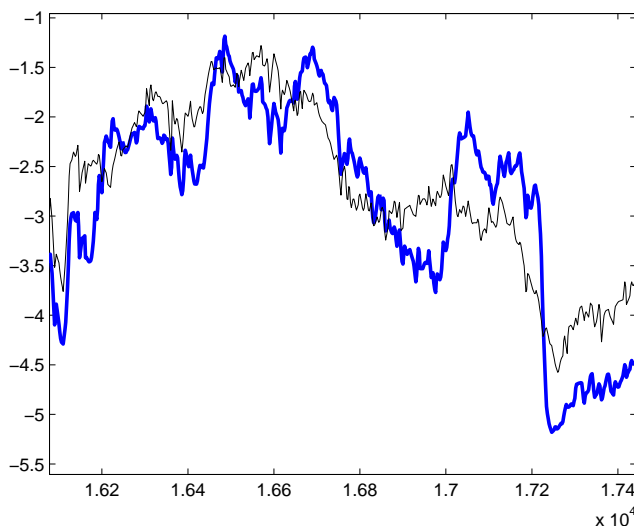


Figura 5: Validación del modelo dinámico en otras condiciones de operación

5 Modelo No Recurrente

Para garantizar, en la práctica, la estabilidad del modelo en sí mismo, suele hacerse uso de modelos no dinámicos, o también llamados Feedforward. En nuestro caso, el resultado de proponer un sistema neuroborroso cuya salida sea la PCM y entradas las componentes principales resultantes del análisis sobre las 56 variables restantes después de un PCA, empeora los resultados, considerándolo inviable en la validación.

Un método alternativo aplicado en este trabajo es usar sistemas borrosos en serie. En la figura 6 se presenta un esquema del sistema propuesto. La idea propuesta consiste en encontrar otro sistema neuroborroso que tenga como entrada el resultado del PCA de las variables menos la PCM y como salidas los datos que se usaron para entrenar las entradas del sistema recurrente.

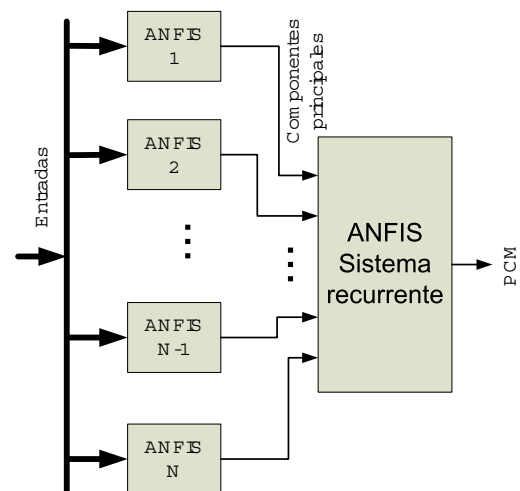


Figura 6: Modelo No-recurrente

Como se observa en la figura 6, en el sistema entran las variables distintas a la PCM y, después de un PCA, se obtienen las componentes principales que recogen el 95% de toda la información del sistema. La evaluación de los sistemas borrosos intermedios, dará como resultado cada una de las componentes principales propias del PCA usado para el modelo recurrente, obteniéndose con ellas, a través del sistema neuroborroso obtenido en la sección anterior, el valor de la PCM. Así pues, a partir de un modelo donde no se tiene en cuenta directamente los valores anteriores de la PCM, se obtiene primero, por estimación, las entradas del modelo recurrente.

La figura 7 describe con un poco más de detalle el esquema del modelo propuesto. Se observa la forma que tienen las distintas funciones de pertenencia de las siete entradas seleccionadas.

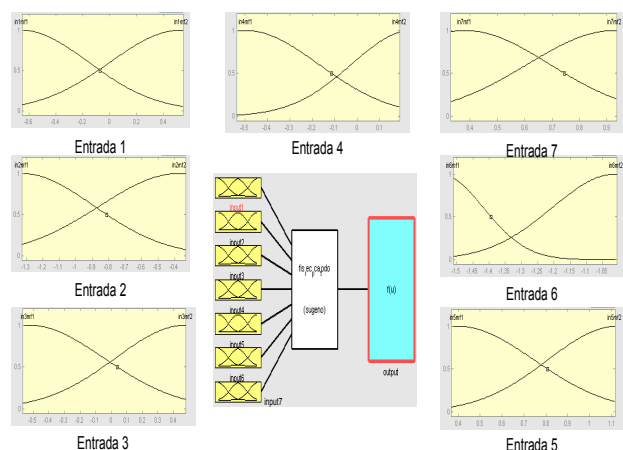


Figura 7: Esquema del modelo

En la figura 8 pueden observarse los resultados de este sistema (el modelo está representado en línea azul).

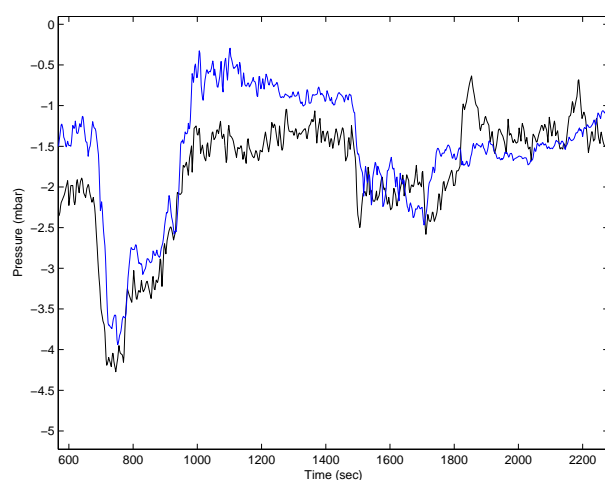


Figura 8: Resultados del modelo No-recurrente

6 Conclusiones

En este artículo se ha presentado modelos neuroborrosos de la presión de la cámara de mezcla de un proceso de producción de ácido sulfúrico, obteniéndose resultados aceptables para simulación y predicción. Se ha hecho uso del método PCA para la reducción del espacio de variables a la entrada. La técnica aquí presentada es un paso más para obtener una metodología adecuada para el modelado neuroborroso, siendo útil éste último al control basado en modelo. También se ha propuesto otro camino para la obtención de modelos no recurrentes basándose en el análisis de componentes principales y en el modelo dinámico obtenido.

Es de destacar que el modelo ha sido validado sobre una planta real, en la cual servirá de ayuda para el diseño y validación posterior de los controladores. La selección de las variables manipulables que tienen mayor influencia en la presión de la cámara de mezcla facilita la tarea de diseño de los controladores.

Agradecimientos

Los autores desean agradecer la financiación de este trabajo al MEC (por su proyecto de investigación DPI2004-07444-C04-01)

Referencias

- [1] A. Trabelsi, F. Lafont, M. Kamoun, G. Enea, "Identification of Nonlinear Multivariable Systems by Adaptative Fuzzy Takagi-Sugeno Model". International Journal of Computational Cognition (<http://www.YangSky.com/yangijcc.htm>). Volume 2. Number 3. Pages 137-153. 2004
- [2] Dallas E. Johnson. "Métodos Multivariados aplicados al análisis de datos". International Thomson Editores S.A. 2000
- [3] Mathworks Inc., "Fuzzy Logic Toolbox User's Guide". Version 2. 3rd printing. *The Mathworks Inc.* 1998
- [4] Jhy-Shyng, Jang, R. "Neuro-Fuzzy Modeling and Control". Proceedings of the IEEE, Vol 83, No. 3, March 1995
- [5] Lee, "Diagonal recurrent neural networks for dynamic systems control," IEEE Trans. Neural Networks, vol.6, pp. 144-156, Jan. 1995.
- [6] K. Kiriakidis, "Fuzzy Model-Based Control of Complex Plants". IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol. 6, NO. 4, Nov.1998
- [7] Ching-Hung Lee, Ching-Cheng Teng, "Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks", IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol.8, no. 4, Aug. 2000

FORMACIÓN ÓPTIMA DE UNA FLOTA DE HELICÓPTEROS AUTÓNOMOS PARA MONITORIZAR BLANCOS MÓVILES TERRESTRES

Adolfo Aumaitre
aumaitre@cartuja.us.es

Aníbal Ollero
aollero@cartuja.us.es

Grupo de Robótica, Visión y Control. Universidad de Sevilla.
Escuela Superior de Ingenieros.
Camino de los Descubrimientos, s/n. 41092. Sevilla.

Resumen

Este artículo presenta un sistema que obtiene la configuración óptima de una flota de helicópteros autónomos que tienen incorporada una cámara para maximizar la información visual obtenida a partir de un objeto móvil que se desplaza sobre la superficie de un terreno.

Palabras Clave: UAV, percepción cooperativa, punto de visión óptimo.

1 INTRODUCCIÓN

Algunas tareas de observación requieren gestionar una configuración espacio-temporal de sensores en un área relativamente amplia. Una aproximación consiste en montar cada sensor en un vehículo aéreo autónomo no tripulado. Concretamente, los helicópteros autónomos son adecuados para muchas aplicaciones debido a su maniobrabilidad. Es más, la capacidad de vuelo estacionario de los helicópteros es muy útil para observación de eventos y tareas de inspección.

Coordinar múltiples UAVs para estas aplicaciones requiere coordinación espacial y temporal. La coordinación espacial distribuye unidades sobre el área que está siendo observada, determinando que un equipo adopte una formación determinada. La coordinación temporal asegura que todos los UAVs actúan en el momento adecuado y suministran sus datos de entrada cuando éstos sean requeridos. La cooperación de equipos busca optimizar la asignación del papel que debe jugar cada vehículo en términos de sus capacidades o restricciones, dirigiendo la formación, coordinación, mantenimiento y dispersión de grupos de vehículos.

Se han desarrollado muchos sistemas basados en flotas de vehículos aéreos autónomos no tripulados en estos últimos años. Algunas aplicaciones representativas que aprovechan las ventajas de los UAVs en tareas de inspección son detección cooperativa usando feromonas digitales [10], búsqueda cooperativa con evitación de colisiones y restricciones de comunicaciones con alcance limitado [34], búsqueda cooperativa de múltiples blancos en un entorno bayesiano [7], búsqueda cooperativa usando modelos teóricos de juegos con restricciones de tiempo de vuelo [30], detección de fuego cooperativa [26] y búsqueda cooperativa usando descomposición de un área en polígonos [12].

En estas aplicaciones, la flota de UAVs persigue dos objetivos que requieren diferentes comportamientos de los UAVs. En localización de blancos debe cubrirse eficientemente un gran espacio de búsqueda y volver a visitar los puntos regularmente, maximizando la probabilidad de detección basándonos en características conocidas del blanco. Cuando un vehículo detecta un blanco, anuncia la posición de éste, y entonces un conjunto de vehículos comienza la observación. En esta fase, los vehículos deben recoger datos visuales del blanco. Para alcanzar este objetivo, es necesario situar los UAVs en algunas coordenadas espaciales en las que obtengan la máxima información visual de dicho blanco. El movimiento del éste podría provocar la reconfiguración de la formación de UAVs. Esta información visual es enviada a una estación base (una antena isotrópica) por medio de las comunicaciones inalámbricas de los UAVs.

Aunque los problemas de gestión de los recursos de sensores, optimización de la localización y formaciones han sido ampliamente estudiados en sistemas multi-agente, hay muy pocos métodos disponibles que puedan ser aplicados al problema de observación presentado en el párrafo anterior.

La mayor parte de aproximaciones de planificación del movimiento del robot suponen que se asigna una ruta a cada agente, así que su principal objetivo consiste en encontrar el camino óptimo entre el origen y el destino como por ejemplo en [15], [17], [13] y [38]. Sin embargo, el problema discutido en este artículo no puede ser considerado un simple problema de origen-destino porque el movimiento de los UAVs depende del movimiento del blanco y el parámetro que debería ser optimizado es la información obtenida a partir de las cámaras y no la longitud del camino de los UAVs.

Otras aplicaciones tradicionales de planificación del movimiento como búsqueda de blancos [29], [27], [23], [6] e interceptación de blancos [33], [18] acaban cuando cada objetivo es visitado. Pero en el escenario discutido aquí, cada UAV debe alcanzar solamente una posición y mantenerla para obtener datos visuales.

Hay métodos cuyos escenarios son similares a los considerados aquí; por ejemplo el paradigma PG2P en [20] supone que los vehículos deben seguir un camino e intenta maximizar la fiabilidad de la percepción a lo largo de ese camino. Sin embargo, este método no es aplicable al problema considerado aquí porque los UAVs no tienen que seguir un camino dado durante la observación, sino que ese camino depende de la posición del blanco.

En [40] el escenario MMT considera el problema de la planificación del movimiento para unos recursos limitados de M sensores móviles en un entorno con N blancos. Las dinámicas de los sensores móviles se modelan como puntos de masa que se mueven a una velocidad constante con un radio de giro limitado. Este método es claramente no adecuado para el problema de los helicópteros ya que sólo trabaja con vehículos en movimiento sin considerar la habilidad de vuelo estacionario de los helicópteros. Es más, el escenario considerado en este artículo tiene un número mayor de UAVs que de blancos y por el contrario los escenarios MMT tienen un número mayor de blancos que de UAVs.

El trabajo de Cortés et al. en [21] es más cercano a la formulación del problema de este artículo, pero aún tiene diferencias considerables. La característica principal que se toma en cuenta para medir la calidad de la observación es la distancia. Otros parámetros a tener en cuenta cuando un blanco está siendo observado con cámaras, como la distorsión, no se mencionan. Es más, este modelo es únicamente aplicable a un escenario 2D llano, que no es el tipo de escenario considerado aquí.

Este artículo se centra en la fase de observación anteriormente descrita. La sección 2 presenta el

problema de encontrar el punto de visión óptimo para un UAV. La sección 3 extiende esa solución para más de un UAV, resolviendo el problema de la coordinación espacial mediante el cálculo de la formación de la flota que obtiene una visión óptima del blanco. La sección 4 trata brevemente del diseño del planificador y de los eventos que pueden repercutir en la ejecución de sus tareas. En la sección 5 se presentan los resultados de la simulación.

2 PUNTO DE VISIÓN ÓPTIMO PARA UN UAV

En la ejecución de una misión se despliegan algunos UAVs y entonces a la flota se le asigna el objetivo de la misión (en este caso, la observación de un determinado blanco obteniendo la máxima información visual). Una vez asignado, los UAVs de la flota tienen que coordinarse para alcanzar el objetivo de la misión.

Debido a la naturaleza de la misión que se trata en este artículo, es necesario juzgar la calidad de la realización de la misión y ajustar su ejecución en relación a esta medida.

Adaptar la ejecución de la misión a la calidad del objetivo de la misión requiere que el planificador sea capaz de calcular la calidad actual y que sea capaz de reconfigurar la formación de la flota para mejorar esta calidad.

En [10] se describen tres aproximaciones al problema de la re-configuración: el modelo local implícito, la evaluación de la idoneidad y la evaluación de la calidad.

La aproximación usada en este trabajo para optimizar la calidad de la misión es la evaluación de la idoneidad. Consiste en construir una función de idoneidad que traduzca un estado determinado de la flota (la posición de los UAVs en el espacio) en un valor de idoneidad que será un número en $[0, 1]$. La función de idoneidad se expresa como:

$$fitness = w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3 \quad (1)$$

donde f_1 es la distancia del UAV al blanco, f_2 es la distorsión de la cámara y f_3 es el ángulo entre el vector que va desde el UAV al blanco y la normal al terreno en el punto donde está el blanco. Estas variables determinan la calidad de la información visual que los UAVs pueden tomar del blanco. Los pesos w_1 , w_2 y w_3 determinan la importancia relativa de cada factor en la solución final. A continuación se analiza cada una de las características mencionadas arriba.

2.1 Distancia

En [20] se introduce el concepto de fiabilidad de la percepción (COP). Esta fiabilidad es proporcional a la resolución de los datos percibidos. Por tanto, esa fiabilidad puede ser expresada en función de la distancia de percepción:

$$COP = f(distance) \quad (2)$$

COP es alta cuando la percepción se ha realizado muy cercana al lugar considerado.

Pero esta aproximación es insuficiente para el problema discutido en este artículo porque los UAVs no tienen que seguir un camino y pueden situarse en cualquier punto en el espacio. El planificador siempre determinaría que los UAVs obtendrían la mejor resolución cuando estuvieran físicamente situados junto al blanco. Ese razonamiento no es adecuado puesto que la distancia óptima entre el UAV y el blanco debe atender a dos criterios, uno de seguridad y otro relativo a la naturaleza de la aplicación. Atendiendo al criterio de seguridad el UAV no debe acercarse tanto al blanco que puedan producirse colisiones con éste o con otros UAVs, o por ejemplo en una aplicación para monitorizar fuego el UAV no debe acercarse tanto a éste como para que peligre su integridad. El criterio de la naturaleza de la aplicación es más difícil de calibrar. Si tenemos una aplicación en la que pretendemos realizar el seguimiento de un blanco, el UAV debe estar lo suficientemente alejado para que sus cámaras puedan encuadrar el blanco completo y no sólo una parte de éste.

De esta forma, se ha determinado que la distancia óptima entre el UAV y el blanco define una esfera alrededor del blanco. Cada punto que pertenezca a la superficie de esta esfera tiene el máximo valor posible de f_i . El valor de f_i disminuirá proporcionalmente con la distancia del UAV a dicha esfera.

2.2 Distorsión de la cámara

En las imágenes que la cámara instalada en el UAV toma del blanco, la distorsión se incrementa con la

distancia de la proyección del objeto en el plano de la imagen al centro de la imagen.

Si se conoce la posición de un punto de la imagen en el mundo real y también se conoce la posición de la cámara a partir de las coordenadas de los UAV y la transformación entre la cámara y las coordenadas del UAV, se puede obtener una relación unívoca entre el mundo real y la posición en pixels.

$$\begin{bmatrix} u' \\ v' \\ s \end{bmatrix} = \begin{bmatrix} \alpha_n & skew & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{bmatrix} \quad (3)$$

donde la primera matriz a la derecha depende únicamente de los parámetros de la cámara, la segunda matriz representa la rotación y la traslación del sistema de referencia del mundo real al sistema de referencia de la cámara y el último vector representa la posición del punto en coordenadas reales homogéneas. Una vez se obtienen u' , v' y s a partir de la ecuación de arriba, las coordenadas en el plano de imagen del blanco pueden ser calculadas como:

$$u = \frac{u'}{s} \quad v = \frac{v'}{s} \quad (4)$$

El siguiente paso consiste en obtener la distorsión, utilizando las siguientes ecuaciones:

$$\tilde{u} = u + (u - u_0) [k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (5)$$

$$\tilde{v} = v + (v - v_0) [k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (6)$$

$$u = u_0 + \alpha x + cy \quad (7)$$

$$v = v_0 + \beta y \quad (8)$$

donde α (distorsión alfa), β (distorsión beta), c , k_1 y k_2 son parámetros de la cámara.

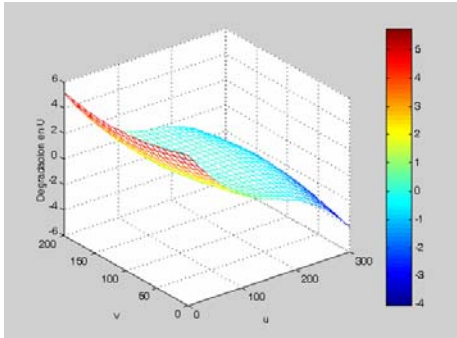


Figura 1.1: Distorsión en u

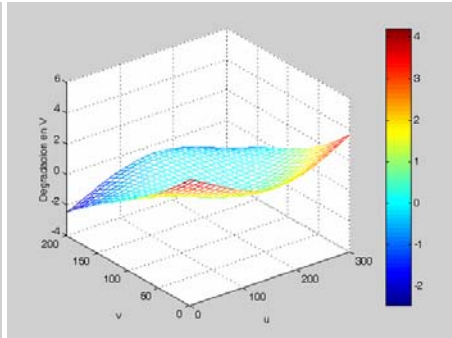


Figura 1.2: Distorsión en v

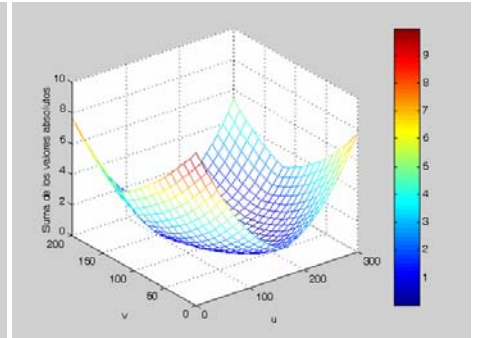


Figura 1.3: Suma de los valores absolutos de la distorsión en u y v

En las figuras 1.1, 1.2 y 1.3 puede verse que la distorsión en el centro de la imagen es 0 y crece si los píxeles están situados en los bordes de la imagen.

Finalmente, una vez se ha calculado la distorsión en todos los puntos, el valor de f_2 en un punto es inversamente proporcional al valor de la distorsión en dicho punto.

2.3 Ángulo

Es el ángulo que forman el vector que va desde el UAV hasta el blanco y el vector normal a la superficie en el punto en que se encuentra el blanco. Para obtener la máxima información visual es necesario que este ángulo sea de 45° . Cuando el ángulo sea de 0° o de 90° , dicha información será mínima.

El método más exacto para obtener el vector normal a la superficie en un punto consiste en calcular las derivadas parciales de la ecuación implícita e la superficie. Pero esta solución no es adecuada en este problema porque el terreno viene dado por una serie de coordenadas con sus alturas, no por una ecuación implícita.

El problema de la poligonización de superficies no implícitas ha sido ampliamente estudiado. Una de las soluciones más aceptadas consiste en modelos de superficies triangulares como en [9], [37] y [5]. En estos casos la obtención del vector normal a la superficie en un punto se reduce a obtener el vector normal al triángulo que contiene el punto.

Sin embargo, esta aproximación vuelve a no tener validez en este artículo. Como se muestra en la figura 2, el vector normal en un punto que está entre dos pendientes distintas será distinto dependiendo de si el triángulo elegido está en la pendiente ascendente o en la pendiente descendente. En la figura 2, el vector \vec{a} es normal a los triángulos marcados con A y el vector \vec{b} es normal a los triángulos marcados con B.

La solución a este problema en los artículos mencionados consiste en incrementar la resolución, dividiendo los triángulos en más triángulos para obtener una aproximación mejor de la superficie. Pero en el problema expuesto en este artículo no es posible obtener más puntos de la superficie para hacer una aproximación mejor. La representación de los terrenos es obtenida a partir de mapas topográficos con muchos kilómetros de longitud y su exactitud no puede ser incrementada.

Suponiendo que el punto en el que el vector normal tiene que ser hallado es (x, y, z) y siendo p_1 la parábola que contiene los puntos $(x + I, y, z)$, (x, y, z)

y $(x - I, y, z)$ y p_2 la parábola que contiene los puntos $(x, y + I, z)$, (x, y, z) y $(x, y - I, z)$, la mejor solución encontrada consiste en obtener la normal al plano tangente a las parábolas p_1 y p_2 .

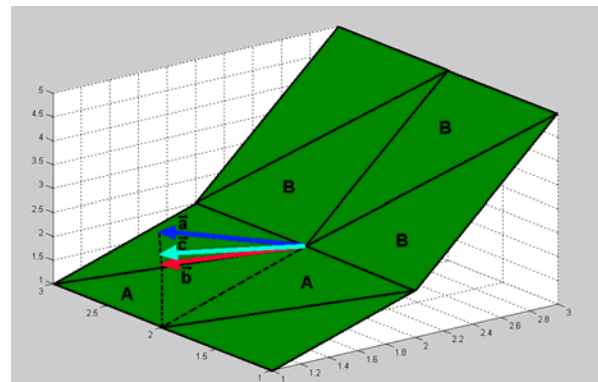


Figura 2: Obtención de la normal en un modelo de superficie triangulada

Este método obtiene una solución más exacta que los modelos de superficie triangulada. En la figura 2 puede verse que el vector \vec{c} que fue obtenido con este método es una combinación lineal de los vectores obtenidos con la poligonización.

Finalmente, el valor de f_3 en cada punto se calcula dependiendo del ángulo entre el vector normal obtenido y el vector que va del blanco al punto donde se levanta la normal. El valor será máximo si el ángulo es de 45° y será mínimo si el ángulo sea 0° o 90° .

3 PUNTOS DE VISIÓN ÓPTIMOS PARA MÚLTIPLES UAVS

Una vez que se ha definido la función que calcula la idoneidad de la posición de un UAV en el espacio, es necesario resolver el problema de la coordinación espacial; es decir, es necesario averiguar cual es la formación de la flota que maximiza la calidad del objetivo de la misión.

La mayoría de trabajos publicados sobre localización multicámara presentan sistemas con cámaras en localizaciones fijas. Estos trabajos se centran en resolver oclusión de objetos [14], en estimar la localización [4], en resolver variaciones de iluminación [35] o en búsqueda multisensor [25]. Existen pocos trabajos que mencionan coordinación espacial en un sistema multicámara. En [10], dicha coordinación se basa en el conocimiento de que volar en una formación rectangular alrededor de un blanco ofrece una imagen de buena calidad.

De acuerdo con el trabajo en [10] y algunos resultados del trabajo en [36] se ha supuesto que la

máxima información se obtiene cuando los UAVs se distribuyen alrededor del blanco de una forma que los vectores que van desde los UAVs hasta el blanco tengan una separación de $360/N$ grados, donde N es el número de UAVs.

Para resolver el problema de distribuir los UAVs de esta forma se creó una matriz M que almacena los ángulos entre un vector de referencia aleatorio y los vectores que van desde todos los puntos del escenario hasta el blanco. Entonces se aplica un algoritmo que en su primera iteración busca todos los puntos que son el origen de un vector cuyo destino es el blanco y que forman con el vector de referencia un ángulo de 0 grados. Se encuentra cual es el que tiene mayor valor de idoneidad y se elige dicho punto. En las siguientes iteraciones se opera igual, pero tomando los puntos que forman con el vector de referencia un ángulo de $(360/NUM_UAVS) \cdot n$ grados, teniendo en cuenta que $n = 1, 2, 3, \dots, NUM_UAVS$. Una vez que se ha obtenido un punto por cada UAV con el máximo valor de idoneidad posible, se suman éstos, obteniendo el valor de idoneidad de la formación.

A continuación se vuelve a ejecutar el algoritmo desde el principio, pero rotando la posición de los UAV un grado, y así sucesivamente hasta que se realizan $360/NUM_UAVS$ rotaciones.

La formación que tenga mayor valor de idoneidad es la que maximiza los parámetros considerados en anteriores apartados.

En este algoritmo es importante el uso de una constante α . Esta constante hace posible que las localizaciones con un buen valor de idoneidad que formen un ángulo cercano a $360/NUM_UAVS$ sean tomados en consideración también (esta desviación máxima permitida es de 2α grados).

El número de iteraciones requerido para cubrir todas las posiciones posibles es $360 / NUM_UAVS$ porque en ese momento todos los UAVs estarán en la posición inicial de los siguientes UAVs en el sentido de las agujas del reloj, lo que significa que esas configuraciones de la flota fueron evaluadas previamente.

4 DIAGRAMA DE ESTADOS DEL PLANIFICADOR

Como se ha indicado antes, el sistema debe ser capaz de reaccionar dinámicamente a cambios en la composición de la flota. Para alcanzar este objetivo es necesario que cada vez que un UAV salga del escenario (ocurra un accidente, vaya a repostar combustible, etc.) o cada vez que se incorpore un

nuevo UAV al escenario, se calcule de nuevo la configuración óptima de la flota.

De forma parecida, el movimiento del blanco provoca una reevaluación de los valores de idoneidad de las localizaciones de los UAVs. Esta reevaluación puede ser evitada cuando la pendiente de esta nueva localización del blanco sea la misma que la pendiente de su localización anterior. En este caso, lo único que hay que hacer es diseñar la trayectoria de los UAVs de forma que sigan exactamente la trayectoria del blanco, de forma que mantengan exactamente las mismas distancias y ángulos a su alrededor.

En la figura 3 se encuentra el diagrama de estados del planificador. En éste se muestran los eventos que pueden producir una reacción por parte del planificador y el nuevo estado al que pasaría éste cuando se diese este cambio.

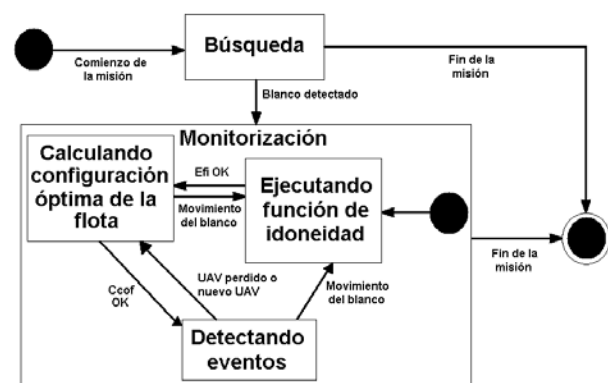


Figura 3: Diagrama de estados del planificador.

5 SIMULACIONES

Para realizar las simulaciones se ha considerado un escenario que consiste en un mapa topográfico de 100×100 puntos. Sobre la superficie de dicho mapa se ha situado un blanco que debe ser monitorizado. En cada ejemplo se ha variado el número de UAVs que se utilizan para observar el blanco y la posición de éste.

El planificador primero calcula las coordenadas en las que los UAVs deben estar situados para obtener la máxima información visual del blanco. Entonces se ejecuta una aplicación que recibe como entradas el mapa topográfico del terreno, las coordenadas de los UAVs, sus orientaciones relativas al blanco y las coordenadas de éste y representa en VRML el terreno y la posición de los helicópteros en relación al blanco, además de los conos de visión que proyectan las cámaras de los helicópteros sobre el blanco.

5.1 Primera simulación

En este ejemplo el blanco se situó en las coordenadas (65, 65, 4) sobre una superficie con una pendiente de 0° . Se estableció que habría dos UAVs disponibles para monitorizar el blanco.

Como puede apreciarse en la figura 4, las posiciones de los UAVs y del blanco conforman un segmento en cuyo centro se encuentra el blanco y con un UAV en cada extremo. Las coordenadas de los UAVs son (72, 77, 11) y (58, 63, 11).

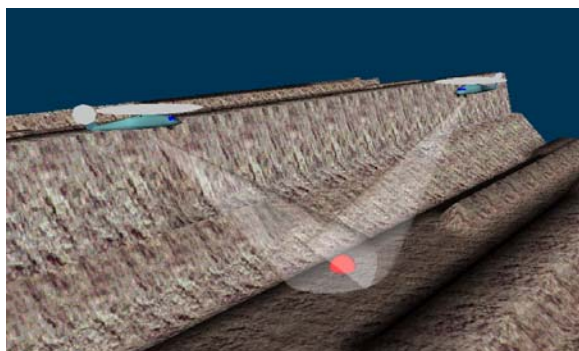


Figura 4: Resultados de la primera simulación.

5.2 Segunda simulación

En esta simulación el blanco fue situado en las coordenadas (17, 17, 3) que, como puede observarse en la figura 5, es un punto situado entre dos pendientes distintas. En esta ocasión se estableció que habría tres UAVs para monitorizar el blanco.

En esta ocasión, el planificador determinó que la formación óptima consistía en una formación en la que cada UAV determina el vértice de un triángulo isósceles (en lugar de un triángulo equilátero, que sería lo adecuado si sólo se tuvieran en cuenta consideraciones geométricas para la determinación de la configuración de la flota). Los UAVs se sitúan en las coordenadas (22, 14, 11), (12, 14, 11) y (15, 19, 13).

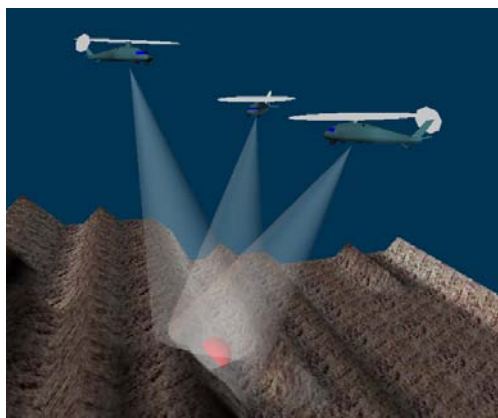


Figura 5: Resultados de la segunda simulación.

5.3 Tercera simulación

Para la última simulación se situó el blanco en las coordenadas (20, 20, 4), en el punto de inflexión entre dos pendientes de la misma inclinación, pero siendo una de ellas de subida y la otra de bajada. Se estableció que habría cuatro UAVs para monitorizar el blanco.

En este último ejemplo, el planificador resolvió que la formación óptima consistía en un cuadrilátero en cuyo interior se encontraba el blanco. Sin embargo, esta figura queda lejos de ser un cuadrado o un rectángulo perfecto, y el blanco no se encuentra en el centro geométrico de la figura que conforman los UAVs, cuyas coordenadas son (25, 12, 8), (13, 16, 10), (15, 19, 13) y (24, 22, 13).

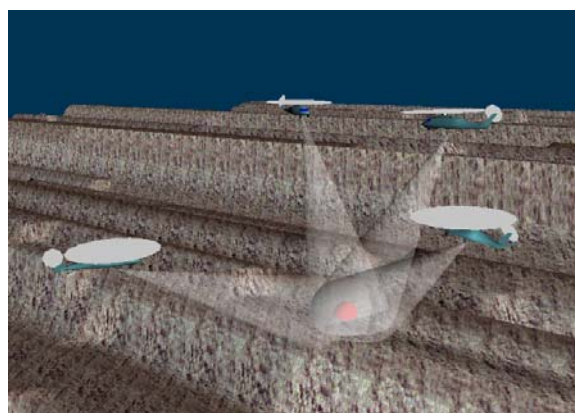


Figura 6: Resultados de la tercera simulación.

Agradecimientos

Este trabajo ha sido subvencionado por una beca de investigación del Ministerio de Educación y Ciencia, referencia AP-2003-4431 y por el proyecto DPI 2005-02293.

Referencias

- [1] A. Okabe, A. Suzuki. Locational optimization problems solved through Voronoi diagrams". European Journal of Operational Research, vol. 98, no. 3, pp. 445-56, 1997.
- [2] A. Okabe, B. Boots, K. Sugihara. Nearest neighbourhood operations with generalized Voronoi diagrams: a review". International Journal of Geographical Information Systems, vol. 8, no 1, pp. 43-71, 1994.
- [3] A. Okabe, B. Boots, K. Sugihara, S. N. Chiu. Spatial Tesselations: Concepts and Applications of Voronoi Diagrams. Wiley Series in Probability and Statistics. John Wiley & Sons, New York, NY, second edition, 2000.

- [4] A. Turolla, L. Marchesotti, C. Regazzoni. Multicamera object tracking in video surveillance applications. IEE Target Tracking 2004: Algorithms and Applications, March 23th-24th, 2004.
- [5] Adrian Hilton, A. J. Stoddart, J. Illingworth, T. Windeatt. Reliable surface reconstruction from multiple range images. In Lecture Notes in Computer Science, vol. 1064, 1996.
- [6] C. Schumacher, P. R. Chandler, S. R. Rasmussen. "Task allocation for wide area search munitions". In Proc. Amer. Control Conf., vol. 3, 2002, pp. 1917-1922.
- [7] El-Mane Wong, Frederic Bourgault, Tomonari Furukawa. Optimal Multi-vehicle Search for Multiple Lost Targets in a Bayesian World. 2005 IEEE International Conference on Robotics and Automation, Barcelona, April, 2005.
- [8] Fernando A. Rodrigues Martins, Pedro Martín Lerones, Jaime Gómez García-Bermejo, E. Zalama Casanova, J. R. Perán González. Digitalización automática de superficies empleando escáneres ópticos. Actas de las XXII Jornadas de Automática en la Industria, Barcelona, sept. 2001.
- [9] Greg Turk, Mark Levoy. Zippered polygon meshes from range images. In Computer Graphics (Siggraph '94), vol. 26, pp. 311-318, 1994.
- [10] H. Van Dyke Parunak, Sven A. Brueckner, James J. Odell. Swarming coordination of multiple UAV's for collaborative sensing. 2nd AIAA Unmanned Unlimited Systems. Technologies and Operations Aerospace land and Sea. Conference and Workshop & Exhibit. September 2003. American Institute of Aeronautics and Astronautics
- [11] H. Yamaguchi and T. Arai. "Distributed and autonomous control method for generating shape of multiple mobile robots group". In IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, Munich, Germany Sept. 1994, pp. 800-807.
- [12] Ivan Maza, Anibal Ollero. Multiple UAV Cooperative Searching Operation Using Polygon Area Decomposition and Efficient Coverage Algorithms. 7th International Symposium on Distributed Autonomous Robotic Systems (7). Num. 7. Toulouse (Francia). Laas-Cnrs. 2004. Pag. 211-220.
- [13] J. Bellingham, A. Richard, J. P. How. Receding horizon control of autonomous aerial vehicle" in Proc. Amer. Control Conf., Anchorage, AK, May 2002, pp. 3741-3746.
- [14] J. Black, T.J. Ellis. Multi Camera Image Tracking. Proceedings of the Second International Workshop on Performance Evaluation of Tracking and Surveillance, December, Kauai, Hawaii, USA, 2001.
- [15] J. D. Boissonnat, A. Cerezo, J. Leblond. Shortest paths of bounded curvature in the plane. In Proc. IEEE Int. Conf. Robot. Autom., 1992, pp. 2315-2320.
- [16] J. P. Desai, J. P. Ostrowski, V. Kumar. Modeling and control of formations of nonholonomic mobile robots". IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pp. 905-8, 2001.
- [17] J. P. Laumond, Robot Motion Planning and Control. Berlin, Germany: Springer-Verlag, 1998.
- [18] J. S. Bellingham, M. Tillerson, M. Alighanbari, J. P. How. Cooperative path planning for multiple UAV's in dynamic and uncertain environments". In Proc. 41st IEEE Int. Conf. Decision Control, vol. 3, 2002, pp. 2816-2822.
- [19] Jeremi Gancet, Gautier Hattenberg, Rachid Alami, Simon Lacroix. Task planning and control for a multi-UAV system: architecture and algorithms. IEEE International Conference on Intelligent Robots and Systems. Edmonton (Canada).
- [20] Jeremi Gancet, Simon Lacroix. PG2P: A perception-guided path planning approach for long range autonomous navigation in unknown natural environments. Proceedings of the 2003 IEEE/RSJ. International Conference on Intelligent Robots and Systems. Las Vegas, Nevada. October 2003.
- [21] Jorge Cortés, Sonia Martínez, Timur Karatas, Francesco Bullo. Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation 20 (2) (2004), 243-255.
- [22] Jules Bloomenthal. Polygonization of implicit surfaces. Computer Aided Geometric Design, vol. 5, pp. 341-355, 1988.
- [23] K. E. Nygard, P. R. Chandler, M. Pachter. Dynamic network flow optimization models

- for air vehicle resource allocation. In Proc. Amer. Control Conf., vol. 3, 2002, pp. 1853-1858.
- [24] K. Sugihara, I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots". Journal of Robotics Systems, vol. 13, no.3, pp. 127-39, 1996.
- [25] L. Marchesotti, A. Turolla, C. Regazzoni. Position estimation and tracking with a network of heterogeneous sensors in video surveillance applications. WIAMIS 2005. 6th International Workshop on Image Analysis for Multimedia Interactive Services.
- [26] Luis Merino, Fernando Caballero, J. R. Martínez de Dios, Aníbal Ollero. Cooperative Fire Detection using Unmanned Aerial Vehicles. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE International Conference on Robotics and Automation. Barcelona, España. IEEE. 2005. Pag. 1896-1901.
- [27] M. Baum, K. Passino. A search theoretic approach to cooperative control for uninhabited air vehicles. In Proc. AIAA GNC Conf. Aug. 2002, paper 2002-4589.
- [28] M. Egerstedt and X. Hu. Formation constrained multi-agent control". IEEE Transactions on Robotics and Automation, vol. 17, no.6, pp. 947-51, 2001.
- [29] M. M. Polycarpou, Y. Yang, K. M. Passino. A cooperative search framework for distributed agents. In Proc. IEEE Int. Symp. Intell. Control, 2001, pp. 1-6.
- [30] P. B. Sujit, D. Ghose. Search by UAVs with flight time constraints using game theoretical models. Proc. of AIAA Guidance, Navigation, and Control Conference, AIAA GNC'2005, 15-18 August 2005, San Francisco, CA, USA.
- [31] P. Tabuada, G. Pappas, P. Lima. Feasible formations of multi-agent systems". In Proc. American Control Conf., Arlington, VA. Jun. 2001, pp 56-61.
- [32] Q. Du, V. Faber, M. Gunzburger. Centroidal Voronoi tessellations: applications and algorithms. SIAM Review, vol. 41, no. 4, pp. 637-676, 1999.
- [33] R. W. Beard, T. W. McLain, M. A. Goodrich, E. P. Anderson. Coordinated target assignment and intercept for unmanned air vehicles". IEEE Trans. Robot. Autom., vol. 18, no. 6, pp. 911-922, Dec. 2002.
- [34] R. W. Beard, T. W. McLain. Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communications Constraints. Proc. IEEE Conf. Decision and Control, Maui, Hawaii, Dec. 2003, pp. 25-30.
- [35] T. Ellis. Multi-camera video surveillance. In Proceedings of the International Carnahan Conference on Security Technology, pp. 228-233. IEEE 2002.
- [36] Tomas Svodova. A Software for Complete Calibration of MultiCamera Systems. Talk given at MIT CSAIL. Jan 25, 2005.
- [37] William E. Lorensen, Harvey E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. In M. C. Stone, editor, Computer Graphics (SIGGRAPH '87 Proceedings), volume 21 (4), pages 163-170, July 1987.
- [38] Y. Guo and L.E. Parker. A distributed and optimal motion planning approach for multiple mobile robots". In Proc. IEEE Int. Conf. Robot. Autom., 2002, pp. 2619-2626.
- [39] Zhengyou Zhang. A flexible new technique for camera calibration. Technical report MSR-TR-98-71. December 1998.
- [40] Zhijun Tang and Ümit Özgüner. Motion Planning for Multitarget Surveillance with Mobile Sensor Agents. IEEE Transactions on Robotics, Vol. 21, no 5. October 2005.

INTERFAZ HÁPTICA MULTIDEDO PARA TAREAS COLABORATIVAS

Carlos González García
cgonzalez@etsii.upm.es

Manuel Ferre Pérez
mferre@etsii.upm.es

Jordi Barrio Gragera
jbarrio@etsii.upm.es

Salvador Cobos Guzmán
cobosalvador@etsii.upm.es

Rafael Aracil Santonja
aracil@etsii.upm.es

Resumen

El presente artículo muestra el desarrollo de una nueva interfaz háptica multidedo empleada para tareas colaborativas de manipulación. Se plantean las consideraciones que se han tenido en cuenta en su diseño y se presentan los distintos modelos cinemáticos de los dispositivos desarrollados. Se presentan asimismo las distintas estrategias para entornos colaborativos cuando se trata de interacciones con objetos.

Palabras Clave: haptics, interfaz multidedo, tareas colaborativas.

1 INTRODUCCIÓN

La creciente necesidad de una mayor fidelidad en las simulaciones y en especial, el deseo de incrementar la sensación de inmersión del usuario dentro de un entorno virtual han incitado la creciente investigación y desarrollo de dispositivos capaces de recrear entornos virtuales. Este aumento de percepción sensorial exige una componente de interactividad que solamente se puede alcanzar mediante dispositivos de tipo táctil.

La utilización del sentido del tacto no se encuentra todavía explorada plenamente, a pesar de que este sentido posee una amplia capacidad para percibir señales dentro de un rango de frecuencias y magnitudes muy amplio.

Cuando se trata de tareas colaborativas, el empleo de sistemas hápticos esta aún menos empleado, y el estudio de cómo se articulan y comportan estos sistemas supone un campo amplio de trabajo en el futuro.

Las interfaces hápticas permiten al usuario tocar, sentir y manipular los objetos simulados en entornos virtuales o sistemas teleoperados. La idea básica de

una interfaz háptica es la de realimentar al usuario las fuerzas generadas en el entorno virtual o remoto como consecuencia de los movimientos realizados por el usuario.

El sistema requiere un dispositivo mecánico, con el que el usuario interactúa con el entorno y que debe generar las fuerzas y momentos que el entorno provoca sobre el usuario. Usualmente van equipados de un dispositivo gráfico, que es el encargado de mostrar el escenario virtual o remoto y proporciona, por tanto, la realimentación visual. Asimismo es preciso tener un software de control que calcule las fuerzas de contacto y genere las señales de consigna para los actuadores.

2 CARACTERÍSTICAS

A la hora de diseñar una interfaz háptica se deben analizar las distintas características técnicas, con el fin de hacer una selección óptima y que está adecuada al uso de la interfaz. Las principales especificaciones técnicas que se deben considerar son:

- Estructura mecánica. Existen básicamente dos tipos de estructuras: las estructuras serie y las paralelas, aunque también son posibles combinaciones híbridas entre ambas arquitecturas
- Tipo de actuadores. Éstos irán adecuados al margen de fuerzas a aplicar y a las posiciones a las que se deba acceder.
- Número de grados de libertad. Son el número de desplazamientos o rotaciones que permite el dispositivo y deben ser acordes a los grados de libertad que el usuario emplea en una tarea.
- Espacio de trabajo. Es el volumen o área dentro del cual el usuario puede posicionar el efector final y dentro del cual el dispositivo puede imponer restricciones a ese movimiento.

- Rango de fuerzas. Es el rango formado por los niveles máximo y sostenido de fuerzas que puede ejercer el dispositivo. Los valores máximos de fuerzas controlable apretando con la mano varían desde 228 N para la mujer hasta los 400 N para el hombre [1]. Sin embargo, si en vez de hacer un agarre fuerte se realiza un agarre de precisión, hay que hacer notar que las máximas fuerzas para los dedos pulgar e índice es de 50 N, mientras que para el resto es de 40 N.

Sin embargo, para que el confort del operador esté dentro de unos niveles de seguridad admisibles, las fuerzas ejercidas por la interfaz no deben exceder del ~15% de los valores máximos anteriormente citados [13].

- Fricción aparente. Las pérdidas por fricción en una interfaz háptica deben ser inferiores a la mínima fuerza o par que el usuario es capaz de percibir mientras interactúa con el entorno virtual. En caso contrario, la interfaz dejaría de ser “transparente”, pues no se podría diferenciar si las fuerzas percibidas por el usuario provienen de la realimentación deseada o de las pérdidas mecánicas del dispositivo en sí. Los valores de fricción aparente deben mantenerse en valores por debajo del 7% de las fuerzas y del 12,7 % de los pares aplicados en la interacción con el entorno virtual [10].
- Rigidez. La rigidez de una interfaz háptica se relaciona íntimamente con la habilidad de la misma para generar restricciones al movimiento del operador dentro del entorno virtual, impidiendo que se penetre dentro de los distintos sólidos virtuales, y permitiendo, de este modo, su inspección y manipulación.

La rigidez máxima de una interfaz depende de la fuerza máxima que pueda desarrollar y del mínimo desplazamiento que detecte. Los estudios demuestran que, en la práctica, una interfaz debe proporcionar una rigidez mínima de 20 N/m para que el operador pueda recorrer adecuadamente una superficie virtual [11].

- Inercia aparente. La inercia aparente es la mínima masa percibida por el usuario cuando mueve el dispositivo háptico a través del espacio. En el caso de que la inercia aparente del dispositivo fuese demasiado alta, podría causar en el usuario una fatiga excesiva. Un estudio piloto [7] indica que son aceptables masas de 50 gramos para operaciones que duren media hora o menos, si bien, estudios recientes recomiendan valores cercanos a los 100 gramos.

- Transparencia. Expresa la capacidad del sistema para seguir el movimiento del usuario rápidamente y sin oposición. Es decir, el usuario no debe percibir ninguna fuerza sobre la mano en tanto que no exista interacción física con el entorno virtual.
- Ancho de banda del dispositivo. Muestra el rango de frecuencias dentro del cual la interfaz háptica es capaz de reflejar fuerzas de forma aceptable.

3 DISPOSITIVOS MULTIDEDO

En cuanto a las interfaces multidedo la mayoría está basada en estructuras tipo guantes [6] (fig. 1) o exoesqueletos [4] (fig. 2) por lo que es el usuario el que debe aguantar el peso de las estructuras y actuadores. En estos casos los actuadores van colocados sobre la palma de la mano o sobre al brazo y las fuerzas que pueden desarrollar sobre cada dedo es de 10 N aproximadamente. Sin embargo, los grados de libertad actuados suelen ser reducidos.



Figura 1: Cybergrasp



Figura 2: Rutgers Master

Últimamente está creciente el interés por interfaces accionadas mediante cables como es el caso de [3,9], con las que se puede acceder a los dedos de manera mejor, aunque existen los problemas de interacciones entre cables y que sólo se actúan un número de grados de libertad sobre el dedo.



Figura 3: SPIDAR

4 INTERFAZ HÁPTICA MULTIDEDO DESARROLLADA

Como alternativa a las estructuras multidedo que existen en la actualidad, presentamos un dispositivo que se conecta a cada dedo y permite la exploración de forma natural y pudiendo realimentar fuerzas en cada dedo con varios grados de libertad (fig. 4).

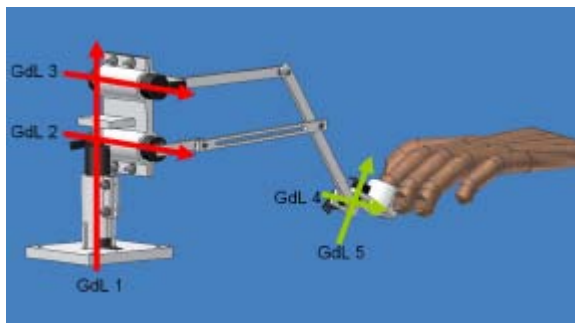


Figura 4: Grados de libertad de la interfaz multidedo

La interfaz está constituida por una estructura serie-paralelo que le permite tener un gran campo de alcance pero que sin embargo le confiere una reducida inercia (fig. 5)

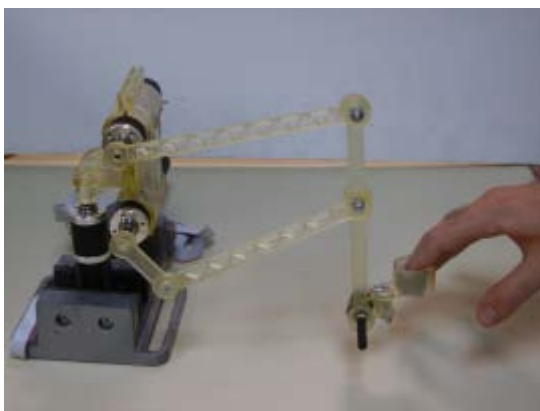


Figura 5: Primer prototipo de la interfaz multidedo

El dispositivo tiene cinco grados de libertad, de los cuales, los tres primeros están actuados. Los dos últimos grados de libertad no están actuados pero sí se mide su posición.

Las longitudes de los eslabones que componen el mecanismo se han seleccionado de tal manera que el campo de alcance sea tal que permita al usuario una gran área de exploración

El “dedal”, donde el usuario inserta su dedo para poder interactuar con el entorno, está provisto de unas tiras resistivas que miden las fuerzas que el usuario está ejerciendo y que sirven para cerrar bucles en fuerza y poder tener un control adecuado.

4.1 CINEMÁTICA DEL DISPOSITIVO

4.1.1 Cinemática directa e inversa

Se presenta a continuación la cinemática de los dispositivos que irán asociados a cada dedo. El primer GdL permite un giro vertical de la mano (aproximadamente correspondiente al movimiento de desviación radio-cubito de la muñeca) mientras que los demás grados de libertad están asociados al movimiento del dedo, propiamente dicho. De entre éstos grados de libertad los más interesantes de estudiar son los dos primeros, que actúan de forma paralela, evitando así la inercia provocada por el movimiento de arrastre de los motores. Una vez resuelta la cinemática de estos dos GdL el estudio de la cadena completa se puede realizar como la de un mecanismo serie con el método de Denavit-Hartenberg, que ha sido explicado ampliamente en la literatura [2] y que por esta razón se omite.

En cuanto a los grados de libertad que forman una estructura paralela, se muestran en la figura 6 los eslabones que la forman y los parámetros dimensionales y angulares que se emplearán.

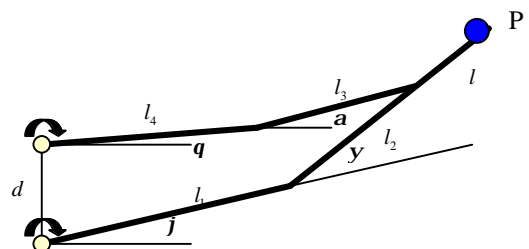


Figura 6: Esquema de los tres primeros GdL

El punto P donde se encuentra el cuarto GdL viene expresado según las siguientes ecuaciones (1):

$$\begin{cases} x = l_1 \cos(j) + l \cos(j + y) \\ y = l_1 \sin(j) + l \sin(j + y) \end{cases} \quad (1)$$

Sin embargo los ángulos conocidos, que son los medidos por los encoders que llevan los motores, son φ y θ , por lo que es necesario hacer una transformación.

El mecanismo paralelo posee dos GdL, por ende, de los cuatro parámetros que definen la posición de los eslabones (φ , θ , α y ψ), únicamente dos son independientes, habiendo por tanto dos ecuaciones de restricción (2) y (3):

$$l_1 \cos(\mathbf{j}) + l_2 \cos(\mathbf{j} + \mathbf{y}) = l_3 \cos(\mathbf{a}) + l_4 \cos(\mathbf{q}) \quad (2)$$

$$l_1 \sin(\mathbf{j}) + l_2 \sin(\mathbf{j} + \mathbf{y}) = d + l_3 \sin(\mathbf{a}) + l_4 \sin(\mathbf{q}) \quad (3)$$

Eliminando α de las ecuaciones anteriores se llega a una ecuación que relaciona φ , θ y ψ (4), que son el conjunto de parámetros conocidos o necesarios para la cinemática.

$$\begin{aligned} l_1^2 + l_2^2 - l_3^2 + l_4^2 + d^2 + 2l_1l_2 \cos(\mathbf{y}) - 2l_1l_4 \cos(\mathbf{j} - \mathbf{q}) \\ - 2l_2l_4 \cos(\mathbf{j} + \mathbf{y} - \mathbf{q}) - 2l_1d \sin(\mathbf{j}) - 2l_2d \sin(\mathbf{j} + \mathbf{y}) \\ + 2l_4d \sin(\mathbf{q}) = 0 \end{aligned} \quad (4)$$

Manipulando adecuadamente la ecuación (4) es posible obtener cualquiera de los parámetros en función de los otros dos. Esto es necesario tanto para el cálculo de la cinemática inversa como la directa. Como ejemplo se muestra como se puede calcular el ángulo ψ en función de los otros dos (5):

$$\mathbf{y} = \arctan(B/A) - \arcsin(b/\sqrt{A^2 + B^2}) \quad (5)$$

donde los parámetros B , A y b son funciones dependientes de las longitudes de los eslabones y de φ y θ (se describen en el anexo I).

4.1.1 Matriz jacobiana

El modelo diferencial, que relaciona velocidades articulares con cartesianas, se formula a través de la matriz jacobiana. Pero la matriz jacobina juega un papel mucho más importante, ya que también relaciona los pares articulares con las fuerzas ejercidas en el efector final (“dedal”) a través de la relación (6).

$$\mathbf{t} = \mathbf{J}^T \cdot \mathbf{F} \quad (6)$$

La importancia de la ecuación (6) radica en su empleo para el cálculo de los pares necesarios en los motores para que el usuario experimente una fuerza determinada.

Para los grados de la estructura paralela la matriz jacobiana viene representada por la siguiente

expresión. Las expresiones de las derivadas parciales vienen descritas en el anexo I.

$$\mathbf{J} = \mathbf{J}_0 \cdot \mathbf{J}_d \quad (7)$$

$$\mathbf{J}_0 = \begin{bmatrix} -l_1 \sin(\mathbf{j}) - l_2 \sin(\mathbf{j} + \mathbf{y}) & -l_2 \sin(\mathbf{j} + \mathbf{y}) \\ l_1 \cos(\mathbf{j}) + l_2 \cos(\mathbf{j} + \mathbf{y}) & l_2 \cos(\mathbf{j} + \mathbf{y}) \end{bmatrix} \quad (8)$$

$$\mathbf{J}_d = \begin{bmatrix} 1 & 0 \\ \frac{\partial \mathbf{y}}{\partial \mathbf{j}} & \frac{\partial \mathbf{y}}{\partial \mathbf{q}} \end{bmatrix} \quad (9)$$

4.1.2 Espacio de trabajo y fuerzas reflejadas

En el proceso de diseño de la interfaz multidedo se ha tenido en cuenta las posibles zonas del espacio a las que el usuario puede acceder. En principio, los puntos a los que puede llegar un dedo son aproximadamente los contenidos en el interior de una semiesfera de diámetro, la longitud de la palma de la mano. Sin embargo, se ha tenido en cuenta que el usuario puede ejercer otros tipos de movimientos relacionados con la muñeca o antebrazo que ayudan a las tareas cooperativas y por tanto el espacio de trabajo que se plantea como objetivo es mayor que el de la palma de la mano. La figura 7 muestra distintas posiciones accesibles para un dedo, pero que son accesibles de igual manera, para la interfaz con todos los dispositivos conectados a cada dedo.

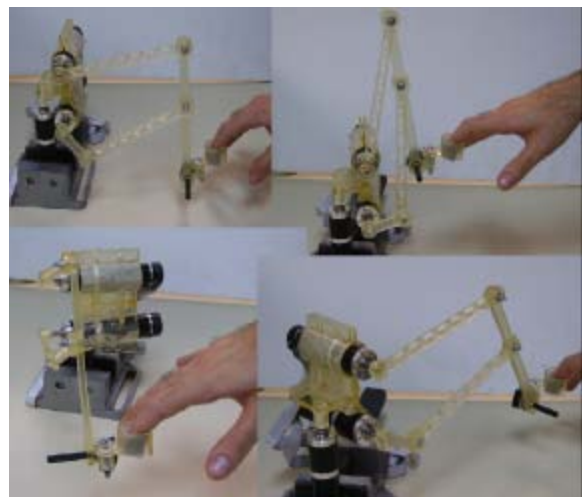


Figura 7: Distintas posiciones accesibles

La figura 8 muestra una sección de las zonas accesibles que permite el dispositivo multidedo cuando se consideran los primeros grados de libertad (El espacio de trabajo total es el obtenido como revolución respecto al eje $x = 0$ de la sección de la figura 8).

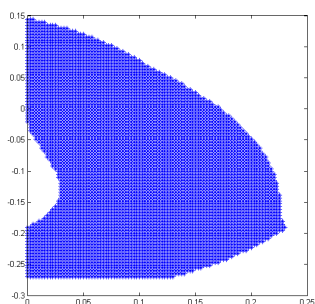


Figura 8: Espacio de trabajo alcanzable

Al igual que es necesario tener en cuenta las zonas accesibles es necesario las fuerzas que el usuario va a poder percibir. Al mismo tiempo es necesario que las longitudes de los eslabones sean tales que hagan que los actuadores no tengan un tamaño excesivo y que se llegue a un compromiso entre las fuerzas que se pueden aplicar y un espacio de trabajo aceptable. Con estas consideraciones las fuerzas que el usuario puede sentir en cada dedo son las que muestra la figura 9.

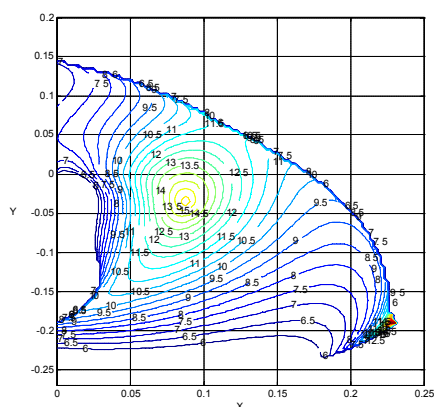


Figura 9: Mínima fuerza de la máxima en cualquier dirección a poder reflejar

5 REALIZACIÓN DE TAREAS COLABORATIVAS

Existen dos configuraciones básicas de agarre: agarre de potencia y agarre de precisión. Con el agarre de potencia se consigue una mayor estabilidad y fuerza, ya que se emplea la palma de la mano, aunque las tareas se realizan con menos destreza. Cuando se emplea el agarre de precisión se utilizan los dedos únicamente y aunque se desarrolle menos fuerza, las tareas se realizan con mayor precisión. Esta habilidad para agarrar un objeto de forma precisa entre los dedos pulgar y índice permite a los humanos y algunos otros primates realizar un amplio rango de complejos y delicados movimientos de la mano. Es por esto que para poder desarrollar tareas colaborativas en las que intervienen varios individuos sea necesario un dispositivo con el que se pueda

interactuar con los dedos y no con una herramienta como habitualmente ocurre en las interfaces hápticas.

En la primera etapa para poder trabajar con entornos cooperativos es conocer el comportamiento humano de la mano y cómo el cerebro es capaz de integrar la información que recibe para generar una respuesta. Para ello la interacción más sencilla es la de los usuarios con los objetos. De las conclusiones obtenidas se podrán plantear estrategias de control para entornos P2P (*persona con persona*) y POP (*persona -objeto-persona*). Esta interfaz permitirá realizar 3 tipos de funciones en la interacción con objetos.

En la fase de grabar (fig. 8) el usuario está conectado a la interfaz, que actúa como dispositivo maestro, mientras que un dispositivo semejante (esclavo) interactúa con el entorno. Los datos generados en esta interacción son recogidos y servirán para recrear un entorno virtual pero con las mismas características que el real.



Figura 10: Acción de grabar

En la fase de reproducción (fig. 9) se plantea una interacción de la persona con el anterior entorno real pero ahora simulado con los parámetros obtenidos del proceso anterior, y previo ajuste de los modelos dinámicos.



Figura 11: Acción de reproducir

La última fase consiste en crear nuevos entornos de los que a priori no se tiene información directa. Para ello se harán uso de los modelos creados en la fase de reproducción (fig. 10).



Figura 12: Acción de crear

Una vez creados los patrones y se hayan modelado las interacciones hombre-objeto, el siguiente paso es la interacción de dos personas distintas sobre el objeto de manera que puedan realizar una tarea conjunta desde el mismo lugar o desde entornos distintos (fig. 13).



Figura 13: Dos personas actuando sobre el mismo objeto

6 CONCLUSIONES

En este artículo se presenta una nueva interfaz multidedo para realizar tareas colaborativas. La interfaz permite manipular objetos haciendo uso de los dedos, pero con un gran espacio de trabajo y con la capacidad de reflejar valores suficientes de fuerzas. La figura 5 muestra el primer prototipo para un dedo.

De igual manera se ha mostrado los modelos cinemáticos necesarios para los distintos cálculos.

Por último, se han planteado estrategias para la interacción colaborativa con objetos (fig. 13).

Anexo I

Parámetros empleados en los cálculos cinemáticos:

$$B = -2 \cdot l_1 \cdot l_2 + 2 \cdot l_2 \cdot l_4 \cdot \cos(\mathbf{j} - \mathbf{q}) - 2 \cdot d \cdot l_2 \cdot \sin(\mathbf{j}) \quad (9)$$

$$A = 2 \cdot l_2 \cdot l_4 \cdot \sin(\mathbf{j} - \mathbf{q}) - 2 \cdot d \cdot l_2 \cdot \cos(\mathbf{j}) \quad (10)$$

$$b = a - 2 \cdot l_1 \cdot l_4 \cdot \cos(\mathbf{j} - \mathbf{q}) - 2 \cdot d \cdot l_1 \cdot \sin(\mathbf{j}) + 2 \cdot d \cdot l_4 \cdot \sin(\mathbf{q}) \quad (11)$$

$$a = l_1^2 + l_2^2 + d^2 + l_3^2 + l_4^2 \quad (12)$$

Derivadas parciales del jacobiano:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{j}} = \frac{-d^2 - l_1^2 - 2 \cdot d \cdot l_4 \cdot \sin(\mathbf{q}) + l_1 \cdot l_4 \cdot \cos(\mathbf{j} - \mathbf{q}) + d \cdot l_1 \cdot \sin(\mathbf{j})}{b - l_2^2 + l_3^2} - \frac{f}{l_2 \cdot (b - l_2^2 + l_3^2)^{\frac{1}{2}}} - \frac{1}{4} \cdot \frac{b}{l_2 \cdot (b - l_2^2 + l_3^2)^{\frac{3}{2}}} \left(\frac{4 - b^2}{l_2^2 \cdot (b - l_2^2 + l_3^2)} \right)^{\frac{1}{2}}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{q}} = \frac{l_4^2 + d \cdot l_4 \cdot \sin(\mathbf{q}) - l_1 \cdot l_4 \cdot \cos(\mathbf{j} - \mathbf{q})}{b - l_2^2 + l_3^2} - \frac{l_4 \cdot (\sin(\mathbf{j} - \mathbf{q}) \cdot l_1 + d \cdot \cos(\mathbf{q}))}{(-(-2 \cdot l_2^2 + 2 \cdot l_2 \cdot l_3 + b) \cdot (-2 \cdot l_2^2 - 2 \cdot l_2 \cdot l_3 + b))^{\frac{1}{2}}} \quad (13 \text{ y } 14)$$

$$f = 2 \cdot l_1 \cdot l_4 \cdot \sin(\mathbf{j} - \mathbf{q}) - 2 \cdot d \cdot l_1 \cdot \cos(\mathbf{j}) \quad (15)$$

Referencias

- [1] An, K-N., Askew, L. y Chao, E., "Biomechanics and Functional Assessment of Upper Extremities", Trends in Ergonomics/Human Factors III, El-sevier Science Publishers, Amsterdam, pp. 573-580, 1986.
- [2] Barrientos A., Peñín L.F., Balaguer C., Aracil R. "Fundamentos de robótica". McGraw-Hill. 1997
- [3] Bouguila L., Murayama J., Hasegawa s., Sato M., "Haptic Interface for Bimanual 6DOF Interaction: SPIDAR-G&G".
- [4] Bouzit M., Burdea G., Popescu G., Boian R. "The Rutgers master II – New design Force-Feedback Glove" IEEE Transactions on Mechatronics, Vol. 7, No. 2, June 2002
- [5] González C., Ferre M., Saltarén R., Barrio J., Aracil R., Ibarra J.M. "Ergonomics and mathematical approach to wire-driven parallel haptic interfaces". Eurohaptics 2006.
- [6] Cybergrasp User Guide. Immersion Co.
- [7] Ellis R.E., Ismaeil O.M. "Design and evaluation of a high performance haptic interface" Robotica vol 4 pp 321-327, 1996.

- [8] Hayward V., Astley O., Cruz-Hernandez M., Grant D., Robles-De-La-Torre G., "Haptic Interfaces and Devices", *Sensor Review*, 24(1), Feb. 2004.
- [9] Kim S., Koike Y., Sato M., "Tension Based 7 DOFs Force Feedback Device: SPIDAR-G" *Transactions on Control, Automation, and Systems Engineering* Vol. 4, No. 1, March, 2002.
- [10] Martín Doñate C., "Interfaces hápticas. Aplicaciones en entornos virtuales". XVI congreso de Ingeniería Gráfica.
- [11] Massie T.H., Salisbury J.K. "The Phantom haptic interface: a device for probing virtual objects" *Proceedings of the ASME Winter Annual Meeting, symposium on Haptic interfaces for virtual environments and teleoperator systems*.1994.
- [12] Schneck D.J., Bronzino J.D. "Biomechanics. Principles and applications". CRC Press.
- [13] Wiker S.F.,Hershkowitz E. Zilk J. "Teleoperator comfort and psychometric stability:criteria for limiting master controller forces of operation and feedback during telemanipulation" *Proc. Nasa conference on space telerobotics vol 1. pp 99-107 Pasadena, CA,USA*.

Detección de sombras en imágenes de satélite de alta resolución Quickbird

G. Ambrosio^a, V. Arévalo^b, J. González^b, J. Valdés^a

^a DECASAT S.L., Parque Tecnológico de Andalucía, Edif. BIC Euronova, 29590 Campanillas, Málaga – {gambrosio, jmvaldes}@decasat.com

^b Dep. de Ingeniería de Sistemas y Automática, Universidad de Málaga, Campus de Teatinos-Complejo Tecnológico s/n, 29071 Málaga - {varevalo, jgonzalez}@ctima.uma.es

Resumen

Las imágenes de alta resolución que proporcionan las últimas misiones de satélites como Quickbird, Ikonos y Orbview han abierto una nueva era en el campo de la teledetección, principalmente en aquellas aplicaciones donde es crucial la extracción de información referente a edificios, calles, vehículos, etc. El incremento de la resolución incide también en que algo inherente a una imagen como son las sombras cobren un especial significado. En este artículo se presentan algunos métodos ya publicados para detectar sombras y se analiza si son adecuados para ser aplicados a imágenes de satélite en color de alta resolución. Basado en este estudio, en este trabajo se presenta un procedimiento que permite la identificación de sombras a partir de invariantes del color y bordes para identificar con exactitud las sombras en imágenes de Quickbird. El método que se describe aquí ha sido probado con éxito sobre imágenes adquiridas bajo diferentes condiciones de iluminación tanto en áreas urbanas como rurales.

Palabras clave: Detección de sombras, índices invariantes del color, imágenes de alta resolución, Quickbird.

1 INTRODUCCIÓN

Las imágenes de alta resolución proporcionadas por las últimas misiones tales como Quickbird, Ikonos, y OrbView han abierto un nuevo rango de aplicaciones en el campo de la teledetección debido a la posibilidad de extraer información detallada de las imágenes. Esas aplicaciones incluyen algo que está comenzando a ser común en los últimos años como la monitorización de desastres naturales (inundaciones, terremotos, etc.) o detección de cambios urbanos, y otras que serán una posibilidad real en los próximos años como la reconstrucción de escenas urbanas, actualización automática de cartografía, inventario urbano, etc.



Figura 1: Sombras en imágenes del Quickbird

El incremento de la resolución espacial en las imágenes de satélites también provoca que algo tan inherente en las imágenes como son las sombras cobren un especial significado por diferentes razones. Por un lado causan la pérdida total o parcial de la información radiométrica en las áreas afectadas y, consecuentemente, provocan una mayor dificultad o incluso la imposibilidad de aplicar procesos de análisis de imágenes tales como la detección y reconocimiento de objetos, la detección de cambios temporales, la reconstrucción de escenas 3D, etc. (ver figura 1). Por otro lado podemos aprovechar la ventaja que proporcionan las sombras considerando la información que facilitan para inferir la estructura 3D de la escena, basada en la posición y la forma de la sombra proyectada, por ejemplo para la detección de edificios, delineación y estimación de alturas [5]. Es clara por lo tanto, la ventaja que la detección precisa de las áreas ensombrecidas en las imágenes de satélite aporta a la restauración de las áreas afectadas y a la generación de información 3D.

La detección de sombras ha recibido en los últimos años gran atención dentro del campo de la visión por computador, aunque no se han desarrollado muchos trabajos destinados a la aplicación de esos resultados a las imágenes de satélite de alta resolución disponibles recientemente. Este artículo describe los principales enfoques que se dan para detectar sombras en imágenes y analiza su idoneidad para ser aplicados a imágenes de satélite en color. Basándonos en este estudio, proponemos un método que explota dos índices invariantes de color en espacios complementarios, así como la información

de bordes para la detección de sombras en imágenes Quickbird (0,6 m./píxel) de forma efectiva y precisa. En este trabajo no se ha dado consideración a la información adicional derivada del azimut del sol o de la altura del terreno (DEM).

De forma muy breve, el método propuesto se basa en la detección de sombras a través de un procedimiento de crecimiento de regiones que fundamentalmente consiste en dos fases:

1. Se seleccionan como semillas de regiones de sombras a pequeños grupos de píxeles que muy probablemente forman parte de sombras. Estos grupos se obtienen del entorno vecino del máximo local en la componente c_3 del espacio de color $c_1c_2c_3$ [4]. Cada área de sombra se caracteriza entonces por una distribución gaussiana de los valores c_3 de los píxeles que se encuentran dentro de esta región.
2. Desde estas semillas se extiende recursivamente la forma de la región de sombra mediante la adición de píxeles adyacentes que son consistentes con la distribución previamente mencionada. Para detectar el área sombreada de la forma más precisa posible, este proceso tiene en cuenta la información sobre la frontera de la región que proporciona el detector de bordes.

Uno de los problemas que se presentan cuando se utiliza la componente c_3 es su inestabilidad para ciertos valores de color, lo que nos lleva a una incorrecta clasificación de píxeles de zonas no sombreadas como pertenecientes a zonas sombreadas (falsos positivos). Tal como se presenta en [9,4], esto ocurre para píxeles con bajo nivel de saturación y para píxeles con valores de intensidad extremos (altos y bajos). Para evitar esto, se tienen en cuenta algunos componentes de los espacios *HSV* y *RGB* que son revisados en las dos fases anteriores. El método presentado ha sido probado con éxito con imágenes adquiridas bajo diferentes condiciones de iluminación tomadas en diferentes estaciones y con diferentes ángulos de elevación de sol sobre áreas tanto de naturaleza urbana como rural.

El resto de este artículo se estructura de la siguiente manera. En la sección 2 revisamos algunos de los métodos más representativos para detectar sombras. En la sección 3 describimos algunos de los espacios de color de especial interés para la detección de sombras. En la sección 4 se describe el método propuesto. En la sección 5 presentamos algunos de los resultados experimentales. Finalmente, destacamos algunas conclusiones y trabajos futuros.

2 UNA REVISIÓN DE LOS MÉTODOS DE DETECCIÓN DE SOMBRAS

En esta sección se presenta una breve introducción a la naturaleza de las sombras (tipos y estructura). También se revisan algunos de los enfoques más representativos en la literatura para detectar sombras en imágenes digitales.

Las sombras se producen cuando los objetos ocultan total o parcialmente la luz directa que proviene de una fuente de iluminación. Las sombras se pueden dividir en dos clases: la *sombra del propio objeto* que consiste en la parte del objeto que no es iluminada por la luz directa, y la *sombra proyectada* por el objeto en la dirección de la fuente de iluminación. La parte de la sombra proyectada donde la luz directa es completamente bloqueada por el objeto se llama *umbra*, mientras que la parte donde la luz directa es parcialmente bloqueada, se llama *penumbra*. Refiérase a [3] para un análisis más profundo de la física de las sombras.

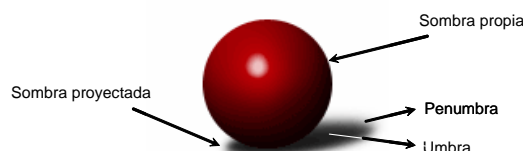


Figura 2: Diferentes tipos de sombras

Cuando se tratan imágenes aéreas o de satélite no se da importancia a las sombras propias de los objetos ni a distinción entre *umbra* y *penumbra*. Así la mayoría de los métodos propuestos en el campo de la teledetección solo tratan con sombras proyectadas siguiendo en general dos enfoques: métodos basados en modelos, o métodos basados en imágenes (basados en las propiedades de las sombras).

2.1 MÉTODOS BASADOS EN MODELOS

En este enfoque se asume que se conoce la geometría 3D y la iluminación de la escena. Esto incluye la localización del sensor o cámara, la dirección de la fuente de luz, y la geometría de los objetos observados, de los cuales se puede derivar un conocimiento a priori de las áreas de sombra. Por ejemplo, podemos considerar regiones poligonales para aproximar las sombras de los edificios u otros elementos urbanos en algunas escenas urbanas poco complejas. Sin embargo, en escenas complejas con gran diversidad de estructuras geométricas, como ocurre habitualmente en las imágenes Quickbird, estos modelos son demasiados restrictivos para proporcionar una buena aproximación. Además, en la mayoría de las aplicaciones, la geometría de la escena y/o las fuentes de luz son desconocidas. Estos hechos limitan la aplicabilidad de este enfoque a

escenas muy restringidas como aquellas utilizadas en [7] y [5].

2.2 MÉTODOS BASADOS EN IMÁGENES

Este enfoque hace uso de ciertas propiedades de las sombras de las imágenes como el color (o intensidad), estructura de la sombra (hipótesis de *umbra* y *penumbra*), bordes, etc., sin ningún tipo de suposición sobre de la estructura de la escena. Más aún, aunque ninguna de esas informaciones estén disponibles, estos métodos pueden ser usados para mejorar el rendimiento en el proceso de detección. Algunas formas habituales de explotar las características de las sombras son:

- El valor de los píxeles en las sombras deben ser bajos en todas las bandas *RGB*. Las sombras son, en general, más oscuras que las regiones de alrededor por lo que tienen que estar delimitadas por bordes perceptibles (límites de sombras) [9,6].
- Las sombras no cambian la textura de la superficie. Las marcas a lo largo de la superficie no tienden a cambiar en los límites de las sombras bajo condiciones generales de observación [6].
- En algunos componentes de colores (o combinaciones de ellos) no se observan cambios con independencia de si la región está sombreada o no, es decir, que son invariantes a las sombras [9,1] (ver sección 3 para más información).

Cuando se dispone de alguna información sobre la escena, esta puede ser usada en combinación con los procedimientos anteriores. Algunos ejemplos de como aplicar esta información son:

- Uno o más bordes de la sombra proyectada están orientados exactamente en la dirección de la luz [8].
- Los tamaños de las sombras dependen de la dirección de la fuente de la luz y de la altura del objeto [11].

Nuestra técnica para la detección de sombras es un método completamente basado en imágenes, puesto que exclusivamente aplicamos propiedades de la imagen. Aunque el azimut del sol y la localización del sensor o cámara están normalmente disponibles en las imágenes de satélite (como por ejemplo en el Quickbird), no hacemos uso de esta información porque en general no es posible la derivación de un modelo preciso del área adquirida (en la mayoría de los casos la geometría 3D de la escena es desconocida o de gran complejidad). En particular, explotamos tanto las componentes invariantes de las sombras como la información de bordes para segmentar las regiones sombreadas. A continuación describimos los diferentes espacios de color y analizamos su idoneidad para ser aplicados a la

detección de sombras en imágenes de satélite de alta resolución en color.

3 ESPACIOS DE COLOR INVARIANTES

Los colores se pueden representar en una variedad de espacios tridimensionales tales como *RGB*, *HSV*, *XYZ*, $l_1l_2l_3$, *YCrCb*, *Lab*, *Luv*, etc. [2]. Cada espacio de color se caracteriza por una serie de propiedades que los hacen especialmente apropiado para aplicaciones específicas. Entre estas propiedades destacamos las características invariantes. Por ejemplo, algunos espacios de color son invariantes a los cambios en las condiciones de la imagen incluyendo la dirección de visualización, la orientación de la superficie del objeto, las condiciones de iluminación, y las sombras. Los espacios de color tradicionales como el *RGB* normalizado (*rgb*), color y saturación (*HSV*) y más recientemente, $c_1c_2c_3$ [4] son representaciones de color que revelan la existencia de alguna propiedad invariante a las sombras. El espacio que mejor comportamiento presenta en este sentido es el $c_1c_2c_3$ el cual ha sido utilizado con éxito por [9] para extraer sombras en imágenes sencillas con pocos objetos de colores únicos y fondo plano (sin textura). Obviamente estas premisas no pueden ser asumidas en imágenes de satélite en color de alta resolución donde las escenas observadas presentan muchas texturas, los objetos tienen colores muy diferentes y la escena, en general, es muy compleja (ver figura 3). Sin embargo para evaluar las limitaciones de este espacio para la detección de sombras en imágenes Quickbird, hemos realizados pruebas sobre un amplio conjunto de imágenes adquiridas bajo diferentes condiciones de iluminación y que cubren tanto áreas urbanas como rurales. El resultado de nuestras pruebas verifica la idoneidad de la componente c_3 para identificar regiones sombreadas, las cuales producen una respuesta más alta que las regiones no sombreadas (ver figura 4). La banda c_3 se calcula a partir de la representación *RGB* mediante la siguiente transformación no lineal:

$$c_3 = \arctan\left(\frac{B}{\max\{R, G\}}\right) \quad (1)$$

Donde *R*, *G*, y *B* son iguales a los componentes rojo, verde y azul respectivamente de cada píxel en la imagen.



Figura 3: Escenas urbanas habituales que ilustran la complejidad de las imágenes de satélite de alta resolución en color (texturas complejas, objetos con diferentes colores, etc.)

A pesar de las prometedoras posibilidades, nuestras pruebas han revelado también los siguientes problemas:

- La banda c_3 es bastante ruidosa lo que puede provocar una clasificación errónea de los píxeles de sombras como no sombreados (verdaderos negativos) además de falta de precisión en los límites de las sombras.
- La ecuación (1) llega a ser inestable debido a los valores bajos de saturación (S) llegando a causar clasificaciones erróneas de píxeles pertenecientes a zonas sin sombra como píxeles sombreados (falsos positivos). Este comportamiento ha sido también referido por [4].
- Los colores cercanos al blanco (altos valores de V) y al azul (altos valores de B) son erróneamente detectados como sombras (falsos positivos). En nuestras imágenes de prueba esto ocurre habitualmente en áreas saturadas y con el agua de las piscinas (como se puede ver en la figura 4).

Para superar los problemas citados en nuestro enfoque hemos incorporado las siguientes dos actuaciones:

- Para minimizar el efecto de ruido, suavizamos la imagen c_3 y utilizamos la imagen gradiente para delimitar con mayor precisión las áreas de sombra.
- Analizamos las componentes S , V (del espacio HSV) y B (del RGB) y no clasificamos un píxel como sombra si se cumple alguno de los problemas mencionados arriba. Esto puede dar lugar a pequeños huecos en una región de sombra que constituyen un precio pequeño a pagar por evitar los falsos positivos, especialmente si pensamos que estos pequeños huecos pueden ser fácilmente rellenados con un filtro morfológico.

A continuación describimos el método desarrollado que está basado en la segmentación de la imagen suavizada c_3 a la par que se tienen en cuenta las consideraciones anteriores.

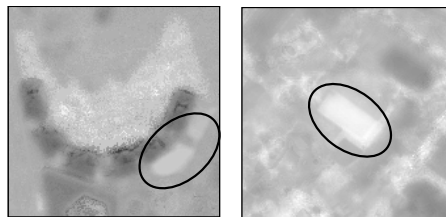


Figure 1: Componentes c_3 correspondientes a las imágenes mostradas en la figura 3. Aquí se muestran marcados los máximos locales producidos por los colores cercanos al blanco (izquierda) y al azul (derecha).

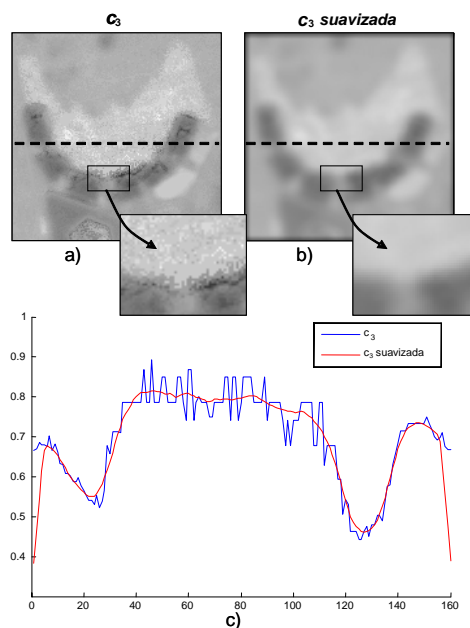


Figura 2: Líneas de barrido horizontal extraídas de (a) la componente c_3 y (b) de la componente c_3 suavizada.

4 DESCRIPCIÓN DEL MÉTODO PROPUESTO

En la figura 6 se muestra un diagrama de bloques del método propuesto que comprende las tres fases que se describen a continuación.

4.1 FASE DE PREPROCESAMIENTO

La entrada al sistema es una imagen RGB de la cual se calculan las siguientes componentes: c_3 , saturación (S) e intensidad (V). La imagen c_3 se convoluciona con un kernel de promediado de 5×5 para minimizar los efectos del ruido (ver figura 5) y la magnitud de gradiente de la imagen de intensidad (V) se calcula mediante la aplicación de un detector Sobel 5×5 (ver figura 7.c).

4.2 FASE DE DETECCIÓN DE SOMBRAS

En esta fase se realiza un proceso genérico de crecimiento de regiones en el que se comienza a partir de un pequeño grupo de píxeles (llamado región semilla) al que se añaden de forma recursiva vecinos que cumplen ciertas reglas de pertenencia.

Esta técnica es particularmente adecuada para nuestro problema ya que en la imagen pueden existir muchas regiones de sombra y cada una de ellas pueden tener perfiles radiométricos muy diferentes (la intensidad o fuerza de la sombra de la sombra puede variar de una zona de la imagen a otra). Obviamente uno de los puntos clave de esta técnica consiste en la correcta colocación de las semillas en la imagen que debe ser de al menos una por región (no pasa nada si hay más de una) y ninguna en píxeles que no son sombra. A continuación se describe nuestra implementación de la técnica en profundidad.

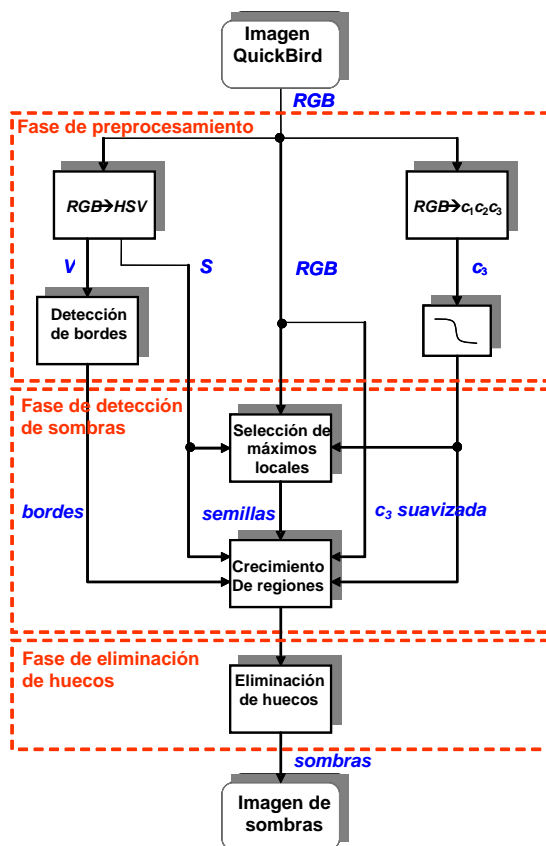


Figura 3: Estructura del sistema propuesto de detección de sombras.

4.2.1 Selección de semillas

Una semilla es una ventana de 9x9 píxeles. Se sitúa en la imagen c_3 suavizada cuando verifica las siguientes condiciones:

1. El centro de la ventana es un máximo local y su 9x9 vecinos tienen valores mayores que la media de la imagen c_3 completa (ver figura 8). El

tamaño de esta ventana constituye el mínimo tamaño permitido para las sombras

2. La media de las componentes azul (B) e intensidad (V) deben ser más bajas que ciertos umbrales T_B y T_V respectivamente, para evitar los problemas que se mencionaron en la sección anterior. De igual manera, la media de la componente de saturación (S) de la ventana de píxeles debe ser más alta que un umbral T_S .
3. Ninguno de los píxeles pertenecientes a la ventana están en otra semilla previa.

En nuestra implementación estos valores han sido empíricamente establecidos a los siguientes valores $T_B=0.65$, $T_V=0.85$ y $T_S=0.02$.

Esas condiciones hacen muy probable que la ventana semilla se corresponda con una sombra. Puesto que en una típica región de sombra se identifican más de una semilla, no nos importa demasiado que no cumplan las premisas que se demandan, lo importante es no generar falsos positivos. Cuando en una región de sombra se sitúa más de una semilla, estas acaban formando una sola región debido al proceso de crecimiento de regiones empleado. Lo único que se podría objetar en este proceso es que tiene más requisitos computacionales que si sólo hubiera una semilla por cada región de sombra, aunque esto no es una cuestión crítica por lo que ha sido obviada en nuestra implementación.

Cada una de las semillas se toma como un prototipo que se caracteriza por la media de la distribución gaussiana $N(c_3)$ de los valores c_3 (ver figura 7). El proceso descrito a continuación descansa sobre esta información para el crecimiento de las semillas sobre las regiones de sombra.

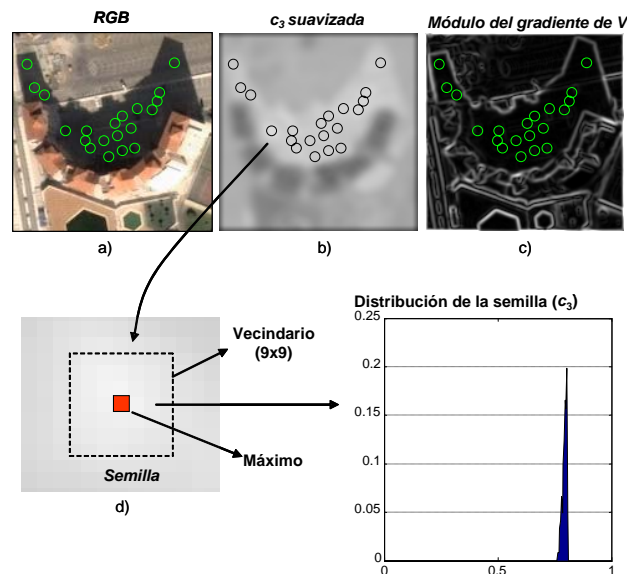


Figura 4: Semillas de sombra identificadas en la componente c_3 suavizada (primera fila). (d) Semilla típica y

la distribución estimada de los píxeles contenidos en el interior del vecindario.

4.2.2 Proceso de crecimiento

Este proceso se ejecuta recursivamente para todos los 8-vecinos de los píxeles situados en los límites de la región de sombra. Comenzando con la ventana semilla, un píxel es clasificado como sombra y añadido a la región si satisface las siguientes condiciones:

- No ha sido previamente añadido a otra región de sombra.
- Tiene que estar por debajo de cierta distancia de Mahalanobis d_0 de la media \hat{c}_3 de la región, es decir, su valor c_3 sigue la distribución gaussiana $N(c_3)$:

$$\frac{|c_3 - \hat{c}_3|}{\sigma} < d_0 \quad (2)$$

donde σ es la desviación estándar de la región.

- Satisface la condición 2 de la sección anterior impuesta para los píxeles semilla.
- La magnitud del gradiente de V está por debajo de un determinado umbral $T_E = 0.25$.

Si el píxel se incorpora la región entonces se actualiza la distribución gaussiana, es decir, se recalculan \hat{c}_3 y σ . El proceso termina cuando ninguno de los píxeles vecinos se añade a la región.

La figura 8 ilustra el proceso de crecimiento en una dimensión. Se muestran distintas líneas de barrido: el módulo del gradiente de la imagen V , y las componentes c_3 y B . Se pueden observar tres ventanas-semilla detectadas y los límites de las sombras (indicados por líneas verticales) donde el proceso de crecimiento de regiones termina debido a la gran magnitud del gradiente.

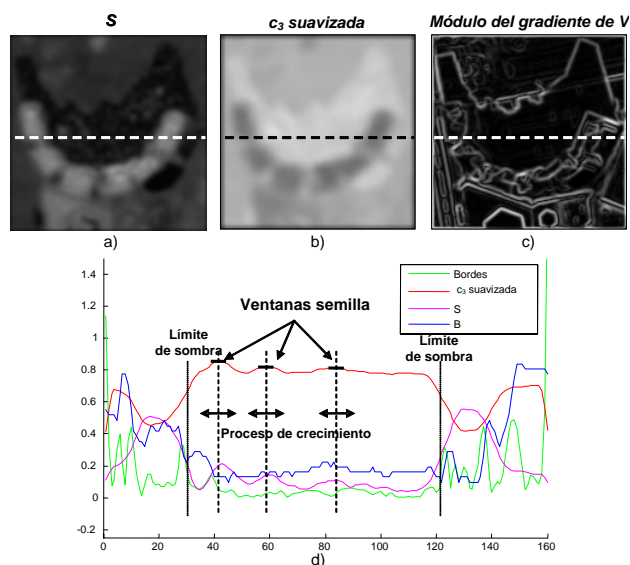


Figura 5: La selección de semillas y las fases de crecimiento se presentan en una línea de barrido horizontal de las imágenes: (a) la componente S , (b) la componente c_3 suavizada y (c) el módulo del gradiente de V . (d) Todas las líneas de barrido trazadas conjuntamente.

4.3 FASE DE RELLENADO DE HUECOS

Después de la fase de crecimiento de regiones aparecen pequeños huecos que se deben a la naturaleza ruidosa de la imagen c_3 . En esta fase aplicamos un par de operaciones morfológicas habituales sobre la imagen binaria de sombras que consisten en una dilatación con un filtro 2x2 seguida de una erosión también 2x2 [10]. Esto rellena huecos de hasta 2 píxeles de anchura.

5 RESULTADOS EXPERIMENTALES

El método propuesto ha sido probado en una gran variedad de imágenes Quickbird adquiridas bajo diferentes condiciones de iluminación (diversas estaciones meteorológicas y diferentes ángulos de elevación del Sol) y sobre áreas urbanas y rurales.

Para mostrar los resultados hemos seleccionado dos porciones de imágenes de la ciudad de Málaga (España): una que cubre un área residencial con edificios y zonas boscosas (figura 9.a), que es de particular interés para comprobar el comportamiento del método en áreas con muchas texturas y con sombras pequeñas e irregulares; y la otra es una típica área urbana (figura 10.a) con edificios altos y una variedad de formas, colores y zonas oscuras que nos permite ilustrar la eficacia del método en presencia de saturaciones y zonas oscuras.

Las figuras 9.b y 10.b muestran las sombras detectadas en las imágenes anteriores para el conjunto de umbrales mencionados en la sección 4. En referencia a estas figuras nos gustaría llamar la atención del lector sobre los siguientes puntos de interés:

- En la figura 9.b, aunque algunas de las sombras pequeñas no han sido detectadas, el método funciona bastante bien dada la dificultad que tiene incluso para un operador humano la distinción entre copas de árboles y sombras. Hemos comprobado que la mayoría de las sombras no detectadas lo fueron por no haberse colocado ninguna semilla en ellas. Esto podría ser corregido de alguna manera pero siempre a costa de aumentar los falsos positivos. Hay que tener en cuenta que con los umbrales seleccionados prácticamente no aparecen falsos positivos en la imagen.
- En la figura 10.b se muestra menos cantidad de regiones de sombra de mayor tamaño que en la

imagen anterior, las cuales están muy bien delimitadas. Las sombras en esta imagen no son fácilmente detectables debido a que se proyectan sobre superficies con mucha textura (ver el edificio de la zona inferior izquierda) y también porque algunos elementos podrían ser erróneamente clasificados como sombras, tal como puede ocurrir por ejemplo en los hexágonos oscuros, el lado saturado de los edificios, etc.

Para cuantificar la efectividad del método hemos comparado dos de las imágenes de sombras detectadas con aquellas manualmente identificadas por un operador (quien ha trazado el contorno con el ratón). Comparando estas imágenes con las correctas hemos calculado los índices de éxitos y fallos que se muestran en la tabla 1.

Tabla 1: El porcentaje de falsos positivos, verdaderos negativos y verdaderos positivos de las dos imágenes de tamaño 2076 x 1024 píxeles de donde fueron extraídas las porciones de los ejemplos aquí mostrados.

	Figura 6.b	Figura 7.b
Falsos positivos	4%	1%
Verdaderos negativos	23%	7%
Verdaderos positivos	77%	93%

6 CONCLUSIONES Y TRABAJO FUTURO

En este artículo hemos presentado un procedimiento para la detección automática de sombras en imágenes de satélite de muy alta resolución. El método propuesto se ha basado en otros trabajos del campo de la visión por computador que fueron desarrollados y aplicados para imágenes de naturaleza muy distinta. En particular, hemos explotado la sensibilidad de los píxeles de sombras a una de las componentes (c_3) de un espacio de color denominado $c_1c_2c_3$. Para superar las limitaciones que la banda c_3 presenta hemos desarrollado un procedimiento de crecimiento de regiones que tiene también en cuenta algunas componentes de otros espacios de color (S , V y B).

Aunque nuestro método necesita de la selección manual de ciertos umbrales, esta operación no tiene que ser muy precisa para la obtención de resultados satisfactorios, aunque esto es algo que queremos evitar en futuros trabajos. Experimentalmente hemos verificado el éxito y efectividad del método para una gran variedad de imágenes en distintas condiciones. Aun así la tasa de verdaderos negativos (sombras reales no detectadas) es todavía alta para algunas aplicaciones como la detección de cambios urbanos.

Como futuro trabajo nos gustaría explotar las áreas de sombra tanto para restaurar la información radiométrica como para inferir información 3D (por ejemplo, la altura de los edificios).

Agradecimientos

Las imágenes QuickBird de DigitalGlobe utilizadas en este estudio han sido distribuidas por Eurimage, SpA. (www.eurimage.com) y proporcionadas por DECASAT Ingeniería S.L., Málaga, España. (www.decasat.com).

Referencias

- [1] Etemadnia, H. y Reza-Alsharif, M. (2003). Automatic image shadow identification using LPF in homomorphic processing system. *Proc. VII Digital Image Computing: Techniques and Applications*, pp. 429-438, Sydney.
- [2] Ford, A. y Roberts, A. (1998). Colour space conversions. *Technical report*, Westminster University, London.
- [3] Funka-Lea, G. y Bajscy, R. (1995). Combining colour and geometry for the active, visual recognition of shadows. *IEEE International Conference on Computer Vision*, pp. 203-209.
- [4] Gevers, T. y Smeulders, A.W.M. (1999). Colour-based object recognition, *Pattern Recognition*, 32, pp. 453-464.
- [5] Irvin B. y McKeown J.R. (1989). Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *IEEE Transactions on System, Man and Cybernetics*, 19(6), pp. 1564-1575.
- [6] Jiang, C. y Ward, M.O. (1994). Shadow segmentation and classification in a constrained environment. *CVGIP: Image Understanding*, 59(2), pp. 213-225.
- [7] Koller, D., Daniilidis, K., y Nagel, H. (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3), pp. 257-281.
- [8] Massalabi, A., He, D.C., Béné, G.B. y Beaudry, É. (2004). Restitution of information under shadow in remote sensing high space resolution images: Application to IKONOS data of Sherbrooke city. *Proc. XX ISPRS Congress*, Istanbul.
- [9] Salvador, E., Cavallaro, A. y Ebrahimi, T. (2001). Shadow identification and classification using invariant colour models. *IEEE International Conference on Acoustic, Speech, and Signal Processing*, 3, pp. 1545-1548, Salt Lake City, Utah.

- [10] Sonka, M., Hlavac, V., y Boyle, R. (1996). *Image Processing, Analysis and Machine Vision*, International Thomson Computer Press, London.
- [11] Stevens, M.R., Pyeatt, L.D., Houlton, D.J. y Goss, M. (1995). Locating shadows in aerial photographs using imprecise elevation data. *Computer Science*

Technical Report CS-95-105, Colorado State University, USA.



Figura 6: Imagen QuickBird de una área residencial que contiene pequeñas edificaciones y zonas boscosas. El número de semillas de sombras identificadas en la imagen es de 189.

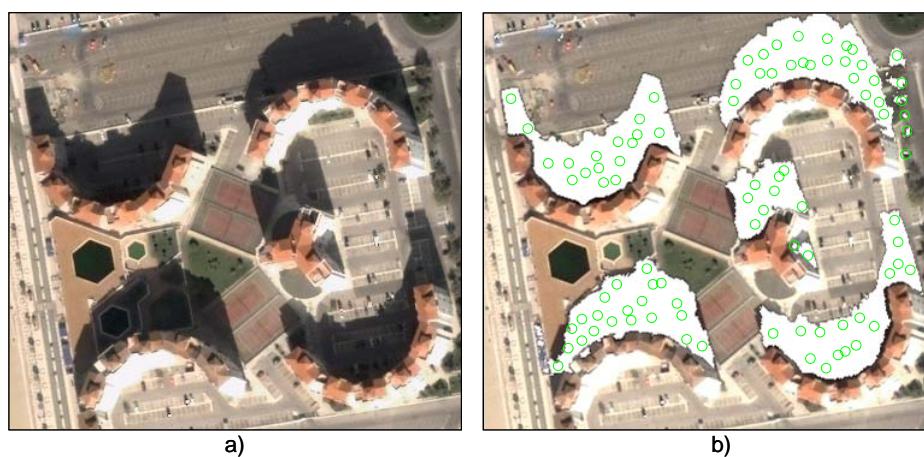


Figura 7: Imagen QuickBird de un área urbana que contiene sombras proyectadas, zonas de saturación y zonas oscuras sobre áreas con mucha textura. El número de semillas de sombra identificadas en esta imagen es de 92.

SEGMENTACIÓN Y ETIQUETADO AUTOMÁTICO DE CABEZAS DE ESPERMATOZOIDES DE VERRACO

Tomás Alonso

Escuela de Ingenierías Industrial e Informática. Universidad de León.
t.alonso@encoweb.com

Enrique Alegre

Escuela de Ingenierías Industrial e Informática. Universidad de León.
enrique.alegre@unileon.es

Lidia Sánchez

Escuela de Ingenierías Industrial e Informática. Universidad de León.
lidia@unileon.es

Fernando Tejerina

Facultad de Veterinaria. Universidad de León.
dsafta@unileon.es

Juan Carlos Domínguez

Facultad de Veterinaria. Universidad de León.
dsajdf@unileon.es

Resumen

Se ha desarrollado una aplicación que permite segmentar y etiquetar cabezas de espermatozoides de verraco como vivas y muertas. Para ello se utilizan pares de imágenes obtenidas de la misma muestra en instantes de tiempo consecutivos. Dichos pares están formados por una imagen bajo fluorescencia y otra en contraste de fases positivo, obtenidas mediante un microscopio epifluorescente Nikon E-600 con y sin iluminación fluorescente. Se han desarrollado dos métodos diferentes para segmentar cada tipo de imagen. La segmentación de la imagen en contraste de fases positivo se basa en el halo blanco que se produce alrededor de la cabeza, utilizando una binarización y un estudio posterior de las regiones obtenidas en función del número de Euler. Para las imágenes en fluorescencia se utilizó un preprocesado con filtro de mediana y filtro de Wiener y una segmentación empleando la transformada máxima extendida con un valor de $H = 30$. La identificación del color en las cabezas para las regiones segmentadas se realizó mediante una relación ad-hoc entre las componentes roja y verde de la imagen en color real. El emparejamiento se realizó con distancia euclídea entre los centroides de las regiones de cada par de imágenes. Los resultados obtenidos son bastante buenos ya que la exactitud del sistema llega al 91,73 %.

Palabras Clave: segmentación, fluorescencia, correspondencia, semen, verraco

1 INTRODUCCIÓN

El área de Ingeniería de Sistemas y Automática lleva varios años colaborando con el área de Reproducción y Obstetricia del Departamento de Sanidad Animal de la Facultad de Veterinaria de la Universidad de León. Esta colaboración se centra en la aplicación de visión, análisis de imagen y reconocimiento de patrones a la evaluación de la calidad del semen de verraco. Para poder realizar estos estudios [2, 5, 6, 7] es necesario disponer de una colección de cabezas de espermatozoides de verraco correctamente segmentadas y etiquetadas. Una de las líneas en las que se está trabajando es en la clasificación automática de las cabezas como pertenecientes a espermatozoides vivos o muertos.

Para ello se estudia la textura de la cabeza en imágenes en escala de grises [5]. Se pretende sustituir las técnicas de laboratorio utilizadas habitualmente que emplean tinciones y luego realizan un conteo manual de las cabezas que presentan dichos estados – vivo o muerto – en una serie de muestras. Estas técnicas clásicas tienen el inconveniente de que son lentas o bien requieren utilizar materiales de laboratorio costosos, como es el caso de los microscopios de fluorescencia.

En este trabajo se presenta una aplicación desarrollada con Matlab que permite realizar la segmentación y el etiquetado automático de las cabezas de espermatozoides de verraco. Se parte de parejas de imágenes tomadas de la misma muestra conteniendo varios espermatozoides, en instantes de

tiempo consecutivos, formadas por una imagen de fluorescencia y otra imagen en escala de grises. En otros trabajos anteriores [1] se había abordado la segmentación de una imagen conteniendo una única cabeza que es precisamente el resultado de este trabajo.

Se realiza la segmentación de las cabezas en la imagen de fluorescencia y de esa manera se obtienen los espermatozoides que hay en la imagen, pudiendo etiquetarlos, en función de su color, como vivos o muertos. Posteriormente se realiza la segmentación en el correspondiente par en escala de grises. Es la misma muestra de la que se ha obtenido una imagen unos segundos después sin utilizar el microscopio de fluorescencia. Los espermatozoides no se encuentran exactamente en la misma posición espacial ya que, al estar en un fluido se produce una deriva de su posición con el tiempo. Se realiza la segmentación de las cabezas de espermatozoide en la imagen en escala de grises y después se estudia la correspondencia entre las cabezas segmentadas en ambas imágenes.

La aplicación desarrollada permite procesar un gran número de imágenes y para cada una de ellas, segmentar, recortar, etiquetar y guardar en disco las cabezas que aparecen en ellas. La segmentación no es trivial al presentarse situaciones diversas, poco controlables, que dificultan esta tarea. Alguno de los problemas que se pueden encontrar son:

- Defectos y ruido en las imágenes
- Objetos parecidos a un espermatozoide pueden ser identificados como tales sin serlo, y viceversa.
- Como ya comentaremos, se trabaja, para cada caso, con una pareja de imágenes. En ocasiones, la correspondencia entre las cabezas de los espermatozoides entre una y otra imagen no es del todo correcta y eso puede introducir más errores. Hemos creado un algoritmo para el emparejamiento de las parejas basado en la distancia euclídea.
- Las cabezas pueden solaparse, es muy difícil distinguir en estos casos cuando se trata de una sola cabeza o varias, o si también ha aparecido ruido.

A continuación se muestran algunas imágenes que presentan dificultades para su correcta segmentación.

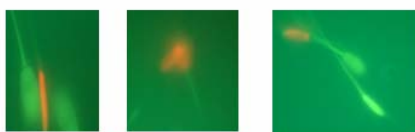


Figura 1: Situaciones difíciles en la clasificación de las cabezas. De izquierda a derecha: cabezas muy

próximas y de canto, cabezas solapadas, espermatozoides solapados y de canto.

2 MATERIALES Y MÉTODOS.

2.1 PREPARACIÓN DE LAS MUESTRAS Y OBTENCIÓN DE LAS IMÁGENES

Para la determinación de la viabilidad de los espermatozoides porcinos se utilizó una tinción combinada de Ioduro de Propidio (IP)-Diacetato de Carboxifluoresceína (DCF); para llevarla a cabo se tomó una muestra de 0.5 mL de una dosis seminal comercial conservada un mínimo de 24 horas a 15 °C; a dicha muestra se le añadió 5 µl de formaldehído al 0,3% en suero salino fisiológico, 10 µl de una solución de IP (0,5 mg/ml) y 10 µl de una solución de DCF (0,46 mg/ ml). Se incubó a 38°C durante 10 minutos, dejando la muestra otros 10 minutos más a temperatura ambiente.



Figura 2: Imagen de fluorescencia.

En la Figura 3 puede verse la imagen en escala de grises correspondiente con la imagen de fluorescencia anterior.



Figura 3: Imagen en contraste de fases positivo.

En la Figura 2 puede verse una muestra de las imágenes de fluorescencia obtenidas. Los espermatozoides que se tiñen y se ven de color rojo están muertos y los teñidos en verde están vivos.

A continuación se realizó la observación en un microscopio epifluorescente Nikon E-600, bajo una excitación de 495 nm a 100x. Una vez focalizados y seleccionados los espermatozoides se procedió a realizar una fotografía bajo fluorescencia de los mismos con una cámara Nikon Coolpix 5000; sin mover la preparación se suprimió la iluminación fluorescente y se abrió el diafragma de luz convencional, obteniéndose una imagen de los mismos espermatozoides en contraste de fase positivo, que fue fotografiada por la cámara previamente descrita. Este proceso se repitió hasta conseguir el número final de imágenes.

2.2 ETAPAS DEL PROCESO

El proceso que sigue nuestro programa, partiendo de la pareja de imágenes es el siguiente:

ETAPA 1.

Segmentación de las imágenes en niveles de grises para obtener las imágenes recortadas de las cabezas de los espermatozoides contenidos en cada muestra.

ETAPA 2.

Segmentación de las imágenes de fluorescencia para obtener, al igual que en la fase anterior, las imágenes recortadas de las cabezas.

ETAPA 3.

Entre las cabezas recortadas obtenidas de ambas segmentaciones se produce un emparejamiento; de esta forma habremos extraído la misma cabeza de las dos imágenes.

Para el emparejamiento usamos la distancia euclídea; dos imágenes recortadas pertenecen a la misma cabeza si la distancia que hay entre sus centroides es la mínima posible.

ETAPA 4.

Clasificación de las cabezas teniendo en cuenta la información de color contenida en la imagen de fluorescencia. Los espermatozoides de color verde están vivos y los que presentan alguna parte roja están dañados o muertos. Se realiza el etiquetado analizando el color de los píxeles que forman la cabeza de los espermatozoides.

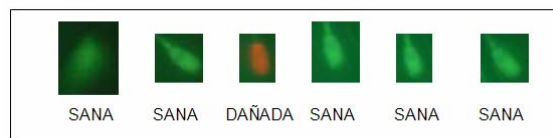


Figura 4: Etiquetado de las cabezas en función de su color.

ETAPA 5.

Se etiquetan las imágenes en escala de grises correspondientes, se realiza un recuento de vivos y muertos cuyos resultados se presentan por pantalla y se graban los recortes con la correspondiente etiqueta.

La aplicación también permite calcular el porcentaje de aciertos comparando los resultados con un conteo realizado visualmente por un ser humano.

2.3 PROCESAMIENTO DE IMÁGENES REALIZADO

2.3.1 Preprocesado y segmentación de las imágenes en escala de grises

La primera parte del tratamiento consiste en una segmentación que permita extraer las cabezas de los espermatozoides en la imagen en escala de grises.

Previamente se preprocesa la imagen para facilitar la segmentación. Para ello se utiliza un filtrado paso bajo utilizando un filtro de media con un kernel de 10x10. Se puede observar el resultado en la figura 5.

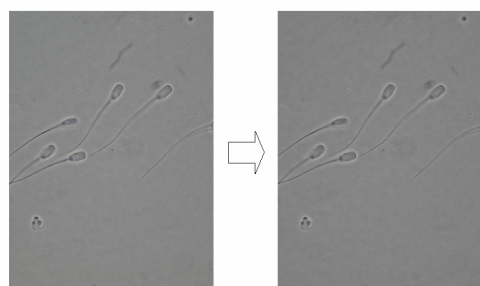


Figura 5. Resultado del filtrado paso bajo.

La segmentación de las cabezas de los espermatozoides se basa en utilizar un efecto en la imagen producido por la obtención de dichas imágenes en el microscopio. Se puede observar un halo blanco que rodea los espermatozoides y principalmente sus cabezas. En base a ello se realiza una binarización con un umbral próximo al blanco, típicamente un nivel de 250, de manera que se extraen las regiones que rodean las cabezas.

Cada cabeza debe ser representada por un único objeto cerrado y no se deben extraer objetos que no correspondan exclusivamente con una cabeza. Para conseguir estos fines además de la umbralización mencionada se utilizan operaciones morfológicas (erosión y dilatación) utilizando kernels en forma de diamante de distintos tamaños.

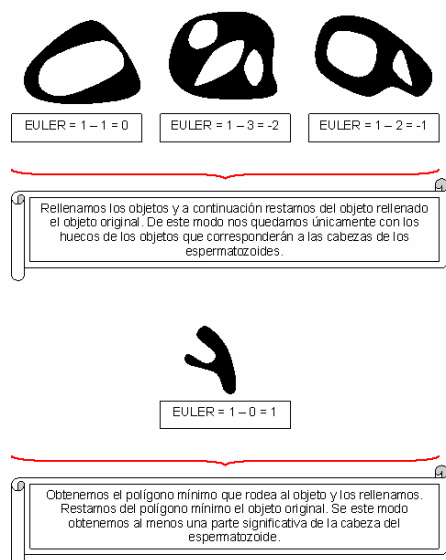


Figura 6. Operaciones morfológicas a realizar en función del número de Euler.

Los resultados de las operaciones realizadas anteriores son muy variadas y las etapas posteriores dependen de cada caso particular. Para discriminar entre una posibilidad y otra se ha utilizado el número de Euler y las operaciones a realizar sobre cada objeto procedente de la umbralización dependerán del número de Euler de dicho objeto.

En la figura 6 puede verse las distintas opciones que aparecen y las operaciones que se realizan en cada caso.

Por último se eliminan los objetos cuyo tamaño sea demasiado pequeño para ser una cabeza de espermatozoide.

2.3.2 Segmentación de las imágenes de fluorescencia

Antes de segmentar las imágenes de fluorescencia se realiza un preprocesado. Como las imágenes son capturadas en color real, lo primero que se hace es convertirlas a escala de grises. Posteriormente se realiza un filtrado lineal adaptativo utilizando una modificación del filtro de Wiener [4]. La imagen resultante es filtrada nuevamente mediante un filtro paso bajo de media con un kernel de tamaño 3.

La segmentación se realiza aplicando la transformada máxima extendida [8] a la imagen con un valor H. En este caso se ha utilizado un valor de H de 30. Esta transformada consiste en obtener la máxima regional de la transformada H-máxima. La máxima regional la forman los píxeles conectados que tienen un valor de intensidad constante y cuyos píxeles del límite externo presentan todos valores inferiores.

El resultado de esta operación es una imagen binaria que se utiliza para enmascarar cada uno de los tres planos, R, G y B de la imagen de color real. Para ello se multiplica cada plano por la imagen binaria obtenida.

Finalmente se vuelve a componer la imagen de color real, se convierte nuevamente a escala de grises y se binariza poniendo a 1 todos los píxeles con un nivel de gris mayor que cero, y dejando a cero aquellos que ya tuvieran ese valor.

2.3.3 Formación de parejas

En cada una de las dos segmentaciones, contraste de fases y fluorescencia, se obtuvieron las regiones que se corresponden con las cabezas de los espermatozoides. Para establecer la correspondencia se busca únicamente asociar los centroides de las regiones utilizando para ello un criterio de mínima distancia.

Aunque la solución propuesta es muy sencilla se comprobó como suficiente en base a los resultados obtenidos.

Es posible que en una y otra segmentación obtengamos distinto número de objetos, si esto ocurre nos quedaremos con el número más pequeño y descartaremos los objetos restantes.

Para formar parejas cogeremos un objeto cualquiera obtenido en la segmentación de una de las imágenes y a partir de su centroide medimos la distancia desde este objeto a todos los objetos obtenidos en la otra imagen. Es decir, asociamos las cabezas que estén más cercanas usando una mínima distancia euclídea.

2.3.4 Clasificación de las cabezas

El estado sano o dañado (o vivos y muertos) de los espermatozoides depende del color de la cabeza. La presencia de alguna zona roja en ella indica que el espermatozoide está dañado o que está muerto.

Se han probado diferentes algoritmos, más o menos complejos, para el reconocimiento de los colores de la cabeza. Al final se ha recurrido a un procedimiento sencillo que consiste en la búsqueda en la cabeza de

los espermatozoides de píxeles que tengan más componentes de un color determinado.

Como ya sabemos una imagen en color está formada por 3 planos o matrices, uno para cada color principal R (red), G (green) y B (blue).

Experimentalmente hemos deducido la siguiente condición de decisión:

Si $G/R > 2$ el píxel es **verde**
Si $G/R < 2$ el píxel es **rojo**

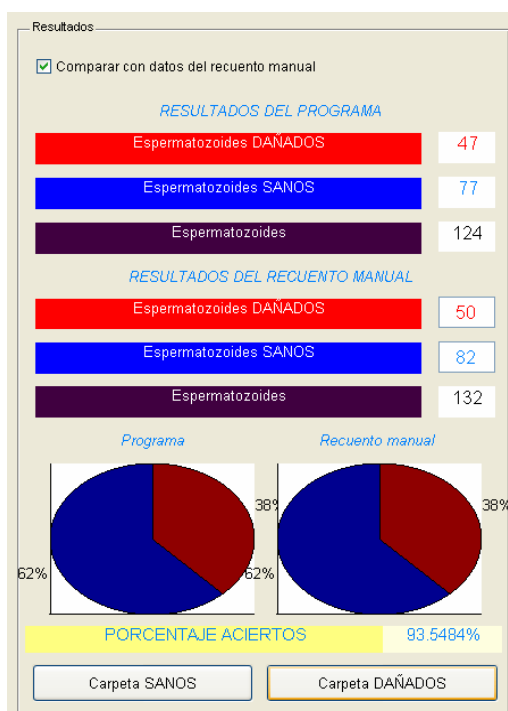


Figura 7: Presentación de los resultados en la pantalla principal del programa.

Es decir, los píxeles verdes son aquellos que tienen, al menos, el doble de componentes verdes que rojas. En caso contrario el píxel es rojo y la cabeza en donde se encuentre ese píxel será identificado como espermatozoide dañado.

Se podría pensar que es demasiado costoso, a nivel computacional, analizar el color de todos los píxeles de la cabeza. Etapas anteriores del proceso han eliminado el fondo y reducido la resolución por tanto el número de píxeles a procesar no es demasiado grande. Además en cuanto se encuentran suficientes píxeles rojos como para identificar el espermatozoide como dañado se da por concluido el análisis de la cabeza y se pasa a la siguiente. En caso contrario, se continúa hasta terminar de revisar todos los píxeles, en busca de regiones rojas que puedan asociarse a un espermatozoide defectuoso.

Este procedimiento de clasificación, aunque sencillo, ha sido el que mejores resultados ha proporcionado y el que menor tiempo computacional ha requerido.

2.3.5. Visualización de los resultados de la clasificación

En la figura 7 puede observarse la forma en la que el programa presenta los resultados finales una vez que ha analizado todas las parejas de imágenes contenidas en una carpeta seleccionada por el usuario.

En una zona de la pantalla principal del programa podrá verse el recuento de espermatozoides sanos, dañados y totales y una comparativa con los datos procedentes de un recuento manual, valores introducidos por el usuario.

En la figura 8 se muestra el resultado de un experimento realizado sobre 37 imágenes.

El programa también almacena las imágenes recortadas y en el nombre del archivo aparece la información de la imagen original de donde fueron obtenidos y también si se trata de una cabeza sana o de una cabeza dañada:

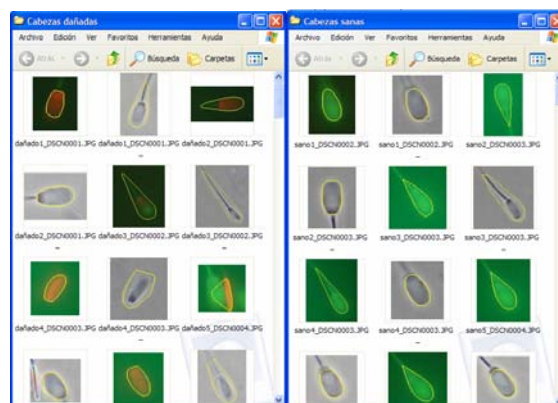


Figura 8. Almacenamiento de las imágenes de las cabezas clasificadas.

3 EXPERIMENTOS Y RESULTADOS

Se ha realizado un experimento para evaluar el funcionamiento de la aplicación sobre un conjunto de imágenes. Hemos utilizado 37 muestras, es decir, 37 parejas de imágenes y es importante resaltar que no hemos realizado ningún tipo de selección de las muestras, todas las muestras disponibles en el momento del experimento fueron analizadas por el programa implementado.

Los resultados fueron los mostrados en las figuras 7 y 8.

De la inspección visual obtuvimos:

50 espermatozoides dañados (rojos)
83 espermatozoides sanos (verdes)
133 espermatozoides

El porcentaje de población de sanos respecto a los totales es de 62,4 %.

El programa obtuvo los siguientes resultados:

47 espermatozoides dañados (rojos)
77 espermatozoides sanos (verdes)
124 espermatozoides

El porcentaje de población de sanos respecto a los totales es según la aplicación del 62,1 %.

Respecto a los aciertos y los fallos que obtuvimos en la clasificación obtenemos los siguientes parámetros que se utilizan para calcular la matriz de confusión [3]:

A Número de espermatozoides dañados identificados como dañados:

$$A = 46$$

B Número de espermatozoides sanos identificados erróneamente:

$$B = 7$$

C Número de espermatozoides dañados identificados erróneamente:

$$C = 4$$

D Número de espermatozoides sanos identificados como sanos:

$$D = 76$$

La matriz de confusión es la siguiente:

		Predicción (aplicación)	
		Negativo (Muerto)	Positivo (Vivo)
Real (Manual)	Negativo	46 (a)	7 (b)
	Positivo	4 (c)	76 (d)

Tabla 1. Matriz de confusión

En los apartados B y C se incluyen los espermatozoides detectados en la inspección visual que no fueron encontrados por el programa. También

hay dos casos en los que el programa detecta espermatozoide cuando en realidad sólo se trata de pequeños defectos en las imágenes, estos dos casos se han repartido por igual entre los apartados B y C.

La matriz de confusión permite calcular algunas medidas para evaluar las prestaciones del método propuesto:

AC Exactitud.

Número de predicciones que son correctas:

$$AC = \frac{a + d}{a + b + c + d} \cdot 100 = 91,73\%$$

TP Ratio de verdaderos positivos.

Porcentaje de espermatozoides sanos identificados como sanos:

$$TP = \frac{d}{c + d} \cdot 100 = 95\%$$

TN Ratio de verdaderos negativos.

Porcentaje de espermatozoides dañados identificados como dañados:

$$TN = \frac{a}{a + b} \cdot 100 = 86,8\%$$

FN Falsos negativos.

Proporción de casos positivos (sanos) clasificados como muertos:

$$FN = \frac{c}{c + d} \cdot 100 = 5\%$$

FP Falsos positivos.

Proporción de casos negativos (muertos) clasificados como vivos:

$$FP = \frac{b}{a + b} \cdot 100 = 13,2\%$$

3. CONCLUSIONES Y TRABAJOS FUTUROS

Los resultados obtenidos en los experimentos han sido bastantes satisfactorios. El objetivo del análisis de muestras era obtener una población de espermatozoides y tanto el recuento automático como la identificación de cabezas como vivos y muertos,

en función del color han dado resultados muy próximos a los obtenidos mediante una inspección visual.

El interés de los experimentos es aún mayor si se tiene en cuenta que hemos trabajado con todo tipo de imágenes, no habiendo preseleccionado las imágenes a utilizar.

Los errores producidos se deben principalmente a las siguientes causas:

- Cabezas demasiado juntas o incluso solapadas. Es difícil separarlas para tratarlas individualmente.
- A veces es difícil saber si algunos objetos que en ocasiones aparecen en las imágenes son espermatozoides u otros objetos o defectos en las imágenes.
- Las fotos presentan distintas tonalidades de colores por lo que la separación en verdes y rojos no es trivial.

Queremos destacar que las etapas más difíciles son las segmentaciones de las imágenes, tanto de la fluorescencia como la de niveles de grises. La mayoría de los errores que han aparecido han sido producidos porque la segmentación no ha sido capaz de separar objetos demasiado juntos o cabezas de un color muy parecido al fondo.

El problema del emparejamiento se ha resuelto correctamente en la mayoría de las ocasiones utilizando la mínima distancia euclídea entre centroides. Hay situaciones en las que, debidas a la deriva que se produce en las cabezas por encontrarse flotando en la suspensión, este emparejamiento produce errores. En trabajos futuros utilizaremos una técnica de correspondencia de formas para mejorar esta asignación.

Indicar finalmente que las líneas de trabajo futuras en las que se trabajará son:

- Segmentación de las imágenes.
- Reducción del tiempo necesario de análisis para cada imagen.
- Posible método de selección de objetos; éstos deberán cumplir unos requisitos (forma, superficie, excentricidad, etc.) para poder ser considerados como cabezas de espermatozoides.
- Mejora del emparejamiento de las cabezas utilizando descriptores de contorno.

Referencias

- [1] Alegre E., Fernández R.A., Sánchez L., Rodríguez V. and Domínguez J.C. (2005). "Digital image segmentation methods for automatic quality evaluation of boar semen", *Reproduction in Domestic Animals*, vol 40.
- [2] Alegre E., Sánchez L., Aláiz R., Domínguez JC (2004). "Utilización de momentos estadísticos y redes neuronales en la clasificación de cabezas de espermatozoides de verraco", XXV Jornadas de Automática, Ciudad Real, Spain, 2004.
- [3] Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, 30, pp. 271–274.
- [4] Lim, Jae S., (1990). *Two-Dimensional Signal and Image Processing*, Englewood Cliffs, NJ, Prentice Hall
- [5] Sánchez L., Alegre E., Fernández R.A. and Domínguez J.C. (2005) "Comparison of texture descriptors in automatic classification of boar semen", *Reproduction in Domestic Animals*, vol 40.
- [6] Sánchez L., Petkov N., Alegre E. (2005). "Classification of boar spermatozoid head images using a model intracellular density distribution", *Computer analysis of images and patterns, Proceedings, Lecture Notes in Computer Science*, 3773, pp 154-160.
- [7] Sánchez L., Petkov N., Alegre E. (2005). "Statistical approach to boar semen classification based on head intensity distribution", *Computer analysis of images and patterns, Proceedings, Lecture Notes in Computer Science*, 3691, pp 88-95.
- [8] Soille, P. (1999). *Morphological Image Analysis: Principles and Applications*, Springer-Verlag,.

Método analítico para el modelado de la calidad proporcionada en servicios multimedia

Diego Próspero*, David López*, Francisco González†, Lidia Sánchez*, Angel Alonso*, and Miguel Sánchez-Urán‡

*Departamento de Ingeniería Eléctrica y Electrónica

Universidad de León, 24071-León, Spain

Email: {diego.prospero,david.lopez,lidia,dieaaa}@unileon.es

†Departamento de Ingeniería Telemática

Universidad Politécnica de Madrid, 28040 -Madrid, Spain

Email: {vidal}@dit.upm.es

‡Departamento de Ingeniería Eléctrica

Universidad Politécnica de Madrid, 28040 -Madrid, Spain

Email: {mas}@ie.upm.es

Abstract—La provisión de servicios multimedia sobre IP está destinada a convertirse en un factor clave en los ingresos de muchos operadores de telecomunicaciones y proveedores de contenidos. Este nuevo paradigma de servicios personalizados, y bajo demanda requiere de nuevas aproximaciones al ciclo de vida completo de la gestión de dichos servicios. Este artículo presenta un nuevo método analítico para determinar la calidad proporcionada en servicios de distribución multimedia sobre redes IP. Se tiene en cuenta la calidad desde un punto de vista del usuario final dado que, en definitiva, esto ayudará a los operadores de red a satisfacer mejor las expectativas de sus clientes. Siendo una aproximación analítica, muestra explícitamente la relación existente entre la calidad proporcionada y las métricas observables en la red para así crear procedimientos operativos para la gestión de servicios.

I. INTRODUCCIÓN

La transmisión de contenidos de audio y video sobre redes IP constituye un cambio en las condiciones iniciales en las que fueron concebidos los algoritmos de codificación. Al principio los codecs de video fueron creados para optimizar la calidad de la imagen decodificada para una tasa de bits fija. Dos condiciones fueron asumidas, un canal de transmisión constante y recursos computacionales suficientes para asegurar un proceso de decodificación en tiempo real. Estas condiciones ya no son válidas y son necesarios nuevos sistemas de codificación capaces de adaptarse a tasas de bits variables y recursos computacionales limitados. El fin principal de cualquier plataforma de distribución multimedia es proporcionar la máxima calidad al usuario sin importar cuales sean las condiciones de la red. Existen varias aproximaciones para superar una potencial degradación en la calidad multimedia proporcionada, la primera, basada en la sobre-provisión reserva ancho de banda para el peor de los escenarios para así asegurarse que hay suficientes recursos disponibles. Otra alternativa se basa en adaptar el ancho de banda necesario a las condiciones existentes en la red, para ello utiliza estrategias de codificación capaces de presentar una respuesta uniforme de decodificación a variaciones imprevistas en la red. Este esquema adaptativo de

codificación, llamado Scalable Video Coding (SVC), se logra mediante métodos, tanto de codificación jerárquica como la llamada Fine Granularity Scalability (FGS).

La codificación jerárquica toma como entrada un grupo de imágenes, las que componen una película, y produce una imagen clave y un conjunto de imágenes no clave. Mientras que la imagen clave ha de ser correctamente decodificada por el cuadro de película en el que es mostrada, el conjunto restante de imágenes que aaden resolución tanto espacial como temporal no tienen porqué ser recibidas necesariamente por el decodificador [1]. Esta característica permite una degradación controlada de la calidad de la imagen, mediante el descarte selectivo de imágenes no-clave, en caso de congestión en la red o escasez de recursos como en dispositivos móviles, por ejemplo.

Las técnicas de codificación FGS como se proponen en [3] constituyen una mejora importante hacia esquemas de codificación más eficientes puesto que reducen significativamente la cantidad de ancho de banda requerido para transmitir contenidos con una resolución de calidad. Para ello, FGS divide imagen sencilla en dos nuevas imágenes: Una actuando como referencia y otra que aade información no crítica extra. Mediante métodos FGS el proceso de codificación es capaz de ajustar la tasa de información al ancho de banda disponible reduciendo el número de bits utilizado para codificar imágenes no críticas.

El codec de video más reciente, recomendación ISO/IEC14496-10, también conocido H.264/AVC [3] propone una nueva arquitectura que, desparejando la codificación de video en dos entidades separadas, es capaz de adaptarse a una gran variedad de medios de transmisión. La primera capa denominada como capa de codificación de video (VCL) está encargado de la representación eficiente de los contenidos mientras que la capa de adaptación a la red (NAL) es la responsable de la transmisión de la información de acuerdo con las especificaciones de la red subyacente. Este planteamiento ha proporcionado esquemas de codificación

con una mejor eficiencia en el ancho de banda como muestran los resultados en [4].

Gracias a los nuevos esquemas de codificación como H.264/AVC y a las tecnologías más eficientes para explotar el ancho de banda disponible en el cable de acceso del cliente, los operadores de red son capaces de progresar hacia servicios multimedia extensivos, interactivos y bajo demanda. La complejidad cada vez mayor de los servicios de nueva generación paradójicamente debido a la cantidad de tecnologías disponibles y a sus propiedades de hechos a medida y bajo demanda, reclaman métodos novedosos para gestionar el ciclo completo de vida del proceso de desarrollo. La provisión de servicios tradicional y las estrategias de gestión de la red ya no son suficientes para cubrir las expectativas de los usuarios. Por un lado la naturaleza en tiempo real de la distribución multimedia requiere una solución diferente que la seguida por la distribución de información basada en TCP en la cual los fallos en la red podrían mantenerse ocultos al usuario. También el transporte de información sobre UDP proporcionando un mecanismo ligero se ha probado que hace a la información bastante sensible a las métricas de la red como el retraso o la pérdida de paquetes. Este informe está organizado como sigue: Primeramente trabajos relacionados y las propuestas de adaptación de la calidad multimedia serán presentadas. En segundo lugar una solución para determinar el comportamiento multimedia en redes IP. En tercer lugar algunos resultados experimentales y finalmente se proporcionan conclusiones y futuro trabajo.

II. TRABAJOS RELACIONADOS

Es numerosa la actividad de investigación orientada hacia la valoración de la calidad percibida por el usuario final. El hecho de ser capaces de inferir cómo el proceso de decodificación se está realizando, sirve para dos propósitos fundamentales, desde el punto de vista del desarrollo se define el banco de pruebas para nuevas propuestas de codificación, por otro lado la capacidad para modelar la percepción del usuario marca la referencia y los objetivos para una arquitectura de gestión multimedia basada en adaptar la calidad a los recursos disponibles.

A. Modelos de percepción de usuarios finales

Los métodos existentes están usualmente basados en procedimientos objetivos o subjetivos. Los objetivos caracterizan la calidad sin intervención humana, por ejemplo la relación señal ruido (PSNR) ha sido ampliamente utilizada [4], [7] como métrica de referencia para la fidelidad de varios esquemas de codificación. Los métodos subjetivos como los mostrados en [8] o en la recomendación ITU-R BT.500 [9] tienen en cuenta la percepción humana para definir métricas de calidad relacionadas con el usuario.

Según la documentación analizada [10] la caracterización de servicios multimedia según la percepción de los usuarios obedece a un comportamiento multidimensional por cuanto intervienen numeros factores en la opinión final acerca de la calidad de los contenidos. Si bien es cierto que tanto la calidad

de las imágenes como el sonido son los aspectos más destacables, el tipo de contenido (deportes, música, noticias) así como características personales (sexo, correctores de visión) intervienen en el proceso de elaboración de opinión.

En [11] se identifica un elevado índice de interacción entre los modos visual y auditivo en la conformación de la opinión final de los telespectadores, invalidando de este modo la aplicación de los modelos unimodales tanto de audio [12], [13] como de vídeo existentes utilizados [14], ya sea en la estimación de servicios de voz así como para la calidad en los procesos de codificación de imágenes.

De manera general los humanos somos más sensibles a la información auditiva que a la visual tal y como demuestran numerosos estudios llevados a cabo [15], [16]. Dicha tendencia se manifiesta de manera especial en determinados contenidos como noticias o vídeos musicales.

En [17] se lleva a cabo una serie de experimentos con usuarios reales que permiten determinar el comportamiento frente a distintas condiciones de visualización de contenidos. Como principales conclusiones de dicho estudio podemos citar:

- i. Los modelos de calidad percibida en contenidos multimedia son multimodales, es decir, es necesario contemplar la respuesta combinada de audio y vídeo.
- ii. La percepción de los usuarios responde a un modelo multiplicativo entre la calidad de vídeo y de audio.
- iii. Los modelos de utilidad son dependientes del tipo de contenido.
- iv. La percepción de los usuarios experimenta una rápida disminución inicial frente a degradaciones de los niveles de calidad visuales y auditivos. Posteriormente la relación entre opinión y calidad, visual y auditiva, obedece a un comportamiento lineal.

Desde el punto de vista de la gestión de un servicio de telecomunicaciones solo aquellos métodos que son capaces de inferir la calidad proporcionada sin decodificar el contenido son válidos, es necesario desarrollar procedimientos que permitan a los operadores de telecomunicaciones gestionar el servicio ofrecido a miles de consumidores mediante procedimientos automatizados que no requieran la visualización de cada contenido distribuido.

Para nuestros propósitos necesitamos definir métricas de calidad y métodos para estimarlas basados en información cuantitativa observable por los operadores de la red IP subyacente.

B. Adaptación de la calidad en servicios multimedia

El trabajo presentado en [18] define dos objetivos, métricas no referenciadas, NR BLOCKINESS y NR Packet Loss rate, estando la última métrica directamente relacionada con el impacto que las pérdidas experimentadas por la red tienen en la calidad de la imagen final decodificada. Debido al contexto específico en el que estas métricas fueron desarrolladas solo son válidas para los esquemas de codificación MPEG2 y no tienen en cuenta otros aspectos importantes, como la calidad del audio.

En [19] varias pruebas subjetivas fueron realizadas para obtener un modelo lineal generalizado que relacione pérdidas de paquetes con las distorsiones observadas, denominadas artefactos, en el contenido decodificado.

Basado en la métrica de calidad no referenciada de una imagen en movimiento definida en [8], un esquema de adaptación multimedia es propuesto en [20]. El llamado esquema de adaptación a la calidad (QOAS) se basa en un agente de monitorización localizado en el domicilio del cliente que periódicamente informa al servidor considerando el estado del contenido proporcionado. Para hacerlo el cliente computa un resultado de calidad al nivel de aplicación (QoDScore) basado en las métricas de red como la tasa de pérdidas el retardo o el parpadeo. De acuerdo con la métrica computada QoDScore el servidor decide en tiempo real si bajar la tasa de bits enviada para adaptarse a la congestión de red existente. De acuerdo con los resultados presentados este método orientado a calidad proporciona una solución escalable para gestionar el punto de vista, la respuesta recibida por el servidor activa una arquitectura autorregulable.

Todos los estudios considerados tienen en cuenta la calidad de la imagen decodificada como el principal banco de pruebas. La percepción del usuario de los nuevos servicios multimedia, como el video bajo demanda (VoD) supone que sean incorporados componentes extra en los futuros esquemas de adaptación de calidad. Aspectos como la calidad de audio, el retardo de acceso de un cuadro aleatorio o el retraso en el cambio de idioma entre otros necesitan ser tenidos en cuenta para cualquier métrica de calidad exhaustiva. Además la mayoría de los métodos están restringidos a escenarios MPEG2 así que no son capaces de utilizar los mecanismos de recuperación de errores como los definidos en H.264/AVC.

3GGP define en su servicio packet switched streaming (PSS) un conjunto de protocolos y funcionalidades que pueden ser utilizados para adaptar sesiones multimedia a los recursos de red disponibles [21] para obtener una experiencia calidad óptima en los usuarios finales. Respecto a la calidad de la experiencia muchas métricas han sido identificadas por 3GGP, particularmente relevantes son: métrica de duración de la corrupción, métrica de la duración de rebuffering, duración de buffering, pérdida de paquetes RPT consecutiva, desviación de la tasa de cuadros y duración del parpadeo.

En nuestra opinión aún se necesitan definir mecanismos para correlacionar la percepción del usuario de los servicios solicitados y las métricas observables. Los proveedores de servicios multimedia necesitan mecanismos de respuesta para regular la entrega de servicios además esta regulación deberá ser realizada en base a procedimientos y tecnologías de gestión de la red existentes.

III. DETERMINACIÓN ANALÍTICA DE MÉTRICAS ORIENTADAS AL CLIENTE

La calidad de la experiencia (QoE) ha sido definida como una extensión de la tradicional calidad del servicio (QoS) en el sentido de que QoE proporciona información respecto a los servicios proporcionados desde un punto de vista del usuario

final [22]. La idea de QoE está directamente relacionada con el concepto tradicional de funciones de utilidad como formas de alto nivel de especificaciones de requisitos. Tanto QoE como las funciones de utilidad permiten marcar grados de deseo para algunos niveles de QoS proporcionados, esta flexibilidad ha pasado a ser de uso práctico en arquitecturas de gestión de autooptimización [23].

Dentro del enfoque multimedia del presente artículo, QoE es definida como la relación causa-efecto entre las métricas de red y los aspectos de calidad como los artefactos de vídeo o cualquier otro parámetro que pueda afectar a la percepción del usuario, esto es para un conjunto de características de calidad dadas $k = 1, \dots, n$, se trata de determinar:

$$QoE_k = f(metric_1, \dots, metric_n) \quad (1)$$

Para determinar la expresión anterior de una forma similar al método presentado en [19] un modelo estadístico será construido en las siguientes secciones, basado en la observación real de imágenes tal como se ven en la pantalla del usuario.

Crear modelos estadísticos relacionando la calidad proporcionada con las métricas de red permitirá a los operadores de red implementar políticas de gestión de servicio adecuadas basadas en informes obtenidos de accesos previos al servicio. Además será posible para ellos gestionar servicios en tiempo real mediante una degradación controlada de la calidad proporcionada basado en un conocimiento precisa de cómo cada métrica de la red está contribuyendo a la percepción del usuario, esto claramente contribuye a reforzar los procesos de gestión de la capacidad como se define en la recomendación ISO20000 [24].

A. Diseo de experimentos y optimización multiobjetivo

El diseo de experimentos (DOE) está definido en [25] como un método genérico estadístico que guía el análisis y diseo de experimentos para encontrar la relación causa-efecto entre respuestas y factores. DOE también puede guiar al experimentador a diseñar experimentos eficientes y a dirigir el análisis de datos para obtener otra información valiosa como la identificación y ordenación de factores importantes. Creemos que DOE es un método adecuado para determinar QoE principalmente por tres razones: Caracteriza el comportamiento de procesos en servicios extremo a extremo, establece una relación explícita entre métricas observables y la calidad obtenida y finalmente proporciona información acerca de la contribución de cada métrica a la variabilidad del proceso de entrega. En resumen, DOE ayuda a deducir la calidad de un servicio multimedia desde el punto de vista del usuario y también las variaciones en la calidad proporcionada en un cuadro determinado.

Generalmente DOE se desarrolla en tres etapas. La inicial está destinada a discriminar factores relevantes que afecten a QoE, en segundo lugar se lleva a cabo un análisis más en profundidad para construir un modelo preciso. Finalmente se tienen en cuenta las preferencias del usuario, para presentar una solución óptima representada mediante una función de preferencia.

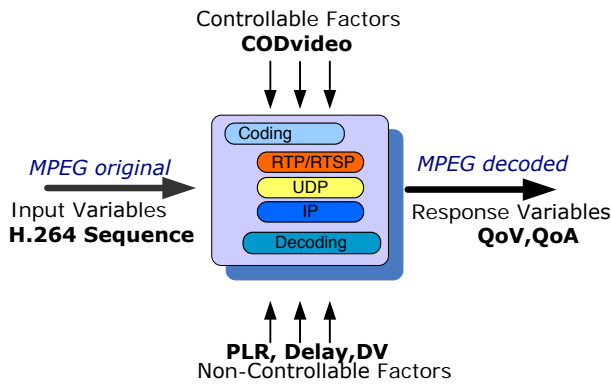


Fig. 1. Modelo de caja negra correspondiente al proceso de codificación, transporte y decodificación de H.264 sobre RTP

IV. RESULTADOS EXPERIMENTALES

El principal objetivo de este experimento es llegar a un diseño cualitativo para el análisis de procesos de transporte y decodificación de contenidos digitales tal como se solicitan por un usuario medio. Para los propósitos de modelado los procesos de distribución y decodificación pueden ser descritos como sistema de caja negra, en los que algunos factores no controlables como la tasa de pérdida de paquetes, el retardo o la variación del retardo interactúan con factores controlables como la tasa de bits de video para producir una respuesta que, en nuestro caso, consiste en una serie de distorsiones de audio y video detectadas, referirse a figura 1. En primer lugar se establecerá la influencia de cada factor en la respuesta de salida (análisis Pareto), determinando posteriormente la contribución de estos factores en la variabilidad del sistema (análisis ANOVA). Para una lectura más detallada acerca del procedimiento los lectores interesados pueden consultar [26].

Los experimentos fueron realizados utilizando equipamiento común para la distribución de contenido multimedia sobre IP [28]. Para analizar el comportamiento del sistema de acuerdo a las variaciones de factores, la red subyacente fue simulada utilizando el software NETEM [27].

Para todas las observaciones, se realizaron pruebas de un solo estímulo dado que refleja la respuesta de un espectador que no tiene acceso al contenido original, de esta manera el espectador decide sobre la calidad basándose en el contenido decodificado una vez que ha atravesado la red.

A. Experimento exploratorio

Como se recoge en [29], [7], [5], los factores más importantes que afectan a la respuesta de los servicios proporcionados sobre redes IP son el retraso de los paquetes, las variaciones en dicho retardo, así como la pérdida y duplicación de paquetes. En el caso de servicios multimedia sobre IP, como codificación de vídeo y algoritmos de encriptación, también han de tenerse en cuenta dichos factores.

Las tablas siguientes muestran tanto los factores considerados como sus valores permitidos. Se ha realizado un experimento factorial completo, 3^4 , y una replicación. Esto es, cuatro

factores con tres valores distintos cada uno haciendo un total de 81×2 observaciones. El procedimiento de cada observación consiste en descargar una película de 180 segundos y contar el número de distorsiones de vídeo y audio que notan los usuarios. Aunque es un experimento que consume mucho tiempo, describe precisamente la opinión del usuario real sobre la calidad de los contenidos.

TABLE I
FACTORES CONSIDERADOS. EXPERIMENTO INICIAL

Variables de respuesta	Descripción
Video Quality - QoE_{video} -	Final outcome of the video decoding. · N_{min} , No distortions were noticed · N_{max} , Max. noticed distortions
Audio Quality - QoA_{audio} -	Final outcome of the audio decoding. · N_{min} , No distortions were noticed · N_{max} , Max. noticed distortions
Factores de red	
Packet Loss Rate -PLR-	Discarded packets rate percentage · N_0 , minimum 0% · N_1 , medium 0.05% · N_2 , maximum 0.1%
Delay -Delay-	end to end delay · N_0 , minimum 0ms. · N_1 , medium 75ms. · N_2 , maximum 150ms.
Delay Variation -DelayV-	end to end delay Variation · N_0 , minimum 0ms. · N_1 , medium 37.5ms. · N_2 , maximum 75ms.
Factores multimedia	
Video coding rate -CODVideo-	Amount of bits to code the video · N_0 , 600kbits/s. · N_1 , 1200kbits/s. · N_2 , 2200kbits/s. · N_3 , 4000kbits/s.

1) 1) *Análisis Pareto, QoE_{video}* : Considerando los resultados de la tabla II, los factores que más afectan a la respuesta obtenida son el retardo y la tasa de codificación de vídeo. Sin embargo, todos los factores son estadísticamente significativos.

TABLE II
FACTORES SIGNIFICATIVOS EN LA VARIABLE DE RESPUESTA QoE_{video}

Effect	Estimate	Std. Error	VIF
Average	11,9211	0,8517	
Packet Loss Rate -PLR-	4,3115	0,8517	1,0
Delay -Delay-	11,9192	1,2633	1,0
Delay Variation -DV-	2,760	1,248	1,0
Video coding rate -CODVideo-	18,6991	1,42218	1,11
Delay*CODVideo	9,303	1,6751	1,0
$CODVideo^2$	-6,438	2,4157	1,11

2) *análisis analysis, QoE_{video}* : Cuando se tiene en cuenta la variabilidad observada, un análisis ANOVA muestra que tanto CODVideo como Delay juegan un papel muy importante, según refleja la tabla III. Hay que señalar, de acuerdo con el valor R-cuadrado, que el modelo se ajusta aproximadamente al comportamiento real de la variable de respuesta QoE_{video} .

3) *Análisis de Pareto, QoE_{audio}* : Según los resultados que se muestran en la tabla IV, la variable de respuesta QoE_{audio} depende del retardo, de la tasa de codificación de vídeo y de la variación del retardo. Hay que indicar que el ratio de pérdida

TABLE III

ANÁLISIS DE LA VARIANZA -ANOVA- PARA LA VARIABLE DE RESPUESTA

 QoE_{Video}

Source	Sum of Sq.	Df	Mean Sq.	F-Ratio	P-Value
PLR	249,27	1	249,279	11,80	0,0009
Delay	1870,0	1	1870,04	89,01	0,0000
DV	102,78	1	102,786	4,89	0,0301
CODVideo	3631,9	1	3631,91	172,87	0,0000
Delay*CODVideo	648,06	1	648,067	30,85	0,0000
$CODVideo^2$	149,21	1	149,216	7,10	0,0094
Total error	1554,6	74	21,0089		

R-squared = 81,573 percent

R-squared (adjusted for d.f.) = 80,079 percent

Standard Error of Est. = 4,58

Mean absolute error = 3,59

Durbin-Watson statistic = 1,508 (P=0,0036)

Lag 1 residual autocorrelation = 0,2281

de paquetes de la métrica no influye significativamente en este caso, debido a la baja exposición de los contenidos de audio codificados a 128Kbits/s a las pérdidas de paquetes de red.

TABLE IV

FACTORES RELEVANTES EN LA VARIABLE DE RESPUESTA QoE_{Audio}

Effect	Estimate	Std. Error	VIF
Average	6,72	1,43	
Delay -Delay-	11,61	1,197	1,03
Delay Variation -DV-	2,6	1,18	1,00
Video coding rate -CODVideo-	13,6	1,35	1,13
Delay*CODVideo	9,59	1,59	1,05

4) *Análisis ANOVA, QoE_{Audio}* : En este caso, el análisis ANOVA muestra que al igual que el CODVideo y el retraso, la variación del retardo juega un papel muy importante, como se refleja en la tabla V.

TABLE V

ANÁLISIS DE LA VARIANZA -ANOVA- PARA LA RESPUESTA QoE_{Audio}

Source	Sum of Sq.	Df	Mean Sq.	F-Ratio	P-Value
Delay	1770,4	1	1770,4	94	0,0000
DV	91,56	1	91,56	4,86	0,0309
CODVideo	1900	1	1900,9	100	0,0000
Delay*CODVideo	682,7	1	682,7	36,25	0,0000
Total error	1242,9	66	18,83		

R-squared = 80,14 percent

R-squared (adjusted for d.f.) = 75,9 percent

Standard Error of Est. = 4,33

Mean absolute error = 2,97

Durbin-Watson statistic = 2,13 (P=0,53)

Lag 1 residual autocorrelation = -0,072

Contrariamente a algunos esquemas de codificación como el MPEG2 transportado sobre MPEG-TS, los resultados muestran que H.264/AVC transportado sobre RTP¹ ofrece una mejora significativa en la resistencia manifestada a la pérdida de paquetes. Los factores CODVideo y Retardo, sin embargo, no afectan a la calidad de vídeo final QoE_{Video} , como se muestra en la figura 2.

¹Real Time Protocol

En el primer diagrama de respuesta del contorno (diagrama de la izquierda), los valores de QoE_{Video} se muestran con respecto a CODVideo y Retardo para un ratio de pérdida de paquetes dado -PLR- del 0%. Hay que indicar que, a pesar de la ausencia de paquetes perdidos, al aumentar el retardo se degrada rápidamente la calidad para casi todos los valores de tasa de codificación -CODVideo-. Este fenómeno empeora cuando la pérdida de paquetes aumenta (diagrama de la derecha) de forma que para pérdidas de paquetes del 1% los ratios de codificación de vídeo mayores que 1200 Kilobits/s no es posible garantizar niveles de calidad aceptables.

Desde el punto de vista de un operador de red, esto implica que cuando los ratios de pérdida de paquetes y de retraso end-to-end son bajos, entonces es posible proporcionar contenidos de alta resolución (es decir, codificación a ratios elevados). Sin embargo, si aumenta la pérdida de paquetes, se deben utilizar ratios de codificación más bajos para evitar la degradación de la calidad.

Por otra parte, se podría decidir codificar a un máximo de 1500 Kilobits/s y trabajar sobre seguro para mantener unos niveles de calidad de vídeo óptimos incluso para rangos amplios de retardo, hasta 150 ms. Hay que sealar que debido a la mayor eficiencia de codificación obtenida por H.264/AVC, la resolución de una película codificada a 1500 Kilobits/s con esta tecnología es equivalente a la resolución producida por una codificación a 3000 Kilobits/s utilizando esquemas anteriores.

B. Análisis de la superficie de respuesta

Como se ha visto anteriormente, las codificaciones de ratio elevado requieren un control estricto de los valores máximos que podían tomar el retraso y el ratio de pérdida de paquetes, ver Figura 2. En este segundo conjunto de observaciones, sólo se considera un rango limitado de valores de dominio, consiguiendo de esta manera obtener más información relativa a algunas áreas específicas de interés y construir un modelo más preciso. El experimento siguiente tiene en cuenta el entorno del rango de codificación $CODVideo \in [2000 Kilobits/s, 4000 Kilobits/s]$, el cual constituye un caso típico de prestación de servicios multimedia sobre redes IP. En todo caso, el mismo procedimiento se aplicará a otras áreas de interés.

Generalmente, el análisis de superficie de respuesta (RSM) es una técnica estadística especializada para construir modelos cuantitativos. El experimento de screening anterior tenía como finalidad caracterizar el comportamiento desde el punto de vista cualitativo. RSM, sin embargo, considera datos más detallados para inferir la respuesta esperada con más precisión, por ejemplo, añadiendo efectos de órdenes mayores (términos cuadráticos). Se puede encontrar más información sobre RSM en [26].

En este caso, el experimento se desarrolló bajo las mismas condiciones, el mismo número de observaciones y factores idénticos. Esta vez, sin embargo, el Retraso máximo fue restringido a 75 ms puesto que valores más altos degradan rápidamente la calidad de vídeo. La metodología propuesta

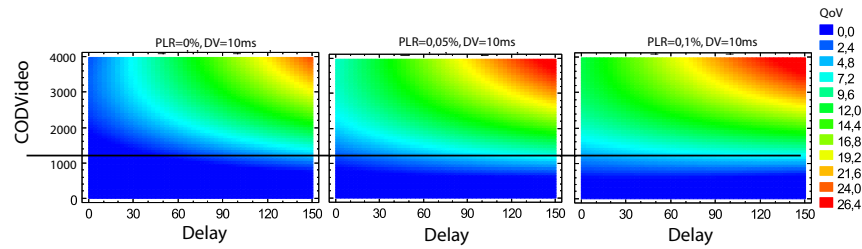


Fig. 2. Evolución de la calidad de vídeo QoE_{video} en función de las pérdidas de paquetes -PLR-

tiene como propósito fundamental definir una expresión analítica para cada variable de respuesta considerada, en este caso para QoE_{Video} y QoE_{Audio} se obtienen las siguientes relaciones:

$$QoE_{Video} = 121,67PLR - 0,399Delay + 0,035DV + \\ + 0,00035CODVideo + 0,002Delay^2 + \\ + 0,002Delay \cdot DV + 0,0001Delay \cdot CODVideo$$

$$QoE_{Audio} = 5,08PLR - 0,11Delay + 0,0074DV - \\ - 0,00008CODVideo + 0,00064Delay^2 + \\ + 0,0006Delay \cdot DV + 0,00003Delay \cdot CODVideo$$

Estas expresiones se derivan de un análisis por regresión donde QoE_{Video} y QoE_{Audio} indican el grado de distorsión de video y audio observado, respectivamente.

Las ecuaciones anteriores establecen las bases para la definición de procedimientos operacionales de gestión de servicios streaming multimedia H.264/AVC sobre RTP/UDP/IP, lo cual es un perfil de tipo-3 como se define en la Recomendación de 2.0 de ISMA [31]. Como el vídeo y el audio se consideran los componentes principales de la calidad percibida, una ventaja importante del método propuesto es la flexibilidad que proporciona DOE para incorporar más aspectos como el retraso rewind/forward, el retraso al cambiar el lenguaje, etc. Para añadir más componentes al modelo de calidad se realizaría de la misma manera.

V. OPTIMIZACIÓN MULTIOBJETIVO

En los apartados anteriores, empleando un enfoque analítico, se ha definido un modelo de calidad de video y audio en multimedia sobre servicios IP. Las expresiones resultantes establecen la tolerancia de ciertas métricas de red para asegurar que se proporciona el servicio adecuado en términos de distorsiones observadas. Hasta el momento no se ha definido qué supone el ser aceptable para los usuarios, es decir, debe el operador proporcionar un servicio 100% libre de distorsiones? Un servicio 80% libre de distorsiones, cubrirá las expectativas del cliente? A qué coste hay que proporcionar el servicio solicitado?

La respuesta parcial a estas preguntas recae, desde el punto de vista matemático, en la llamada programación multiobjetivo

entre otros métodos de programación no lineales. En este trabajo, se necesitan encontrar soluciones óptimas para las variables de respuesta - QoE_{Audio} , QoE_{Video} - dadas ciertas restricciones en el nivel de degradación que los clientes están dispuestos a soportar. Formalmente, el problema se podría describir de la siguiente manera:

Para cada respuesta QoE_i una función de preferencia $d_i(QoE_i)$ asigna un número real entre 0 y 1 a los posibles valores de QoE_i . $d_i(QoE_i) = 0$ representa un valor no deseable completamente y $d_i(QoE_i) = 1$ representa un valor de respuesta ideal. A continuación, se combinan los valores deseados individuales utilizando la media geométrica:

$$D = \{d_1(QoE_1) \cdot d_2(QoE_2) \cdots d_n(QoE_n)\}^{1/n} \quad (2)$$

En la literatura existen diversas propuestas para construir modelos de preferencia. En este caso hemos optado por las propuestas en [32] principalmente por la flexibilidad a la hora de seleccionar la respuesta de los usuarios en función del tipo de contenido presentado (noticias, deportes o cine).

Si fijamos un máximo de 15 distorsiones visuales admisibles durante un período de 2 minutos así como un número máximo de 3 distorsiones auditivas obtenemos la serie de funciones representadas en las figuras 3.

Se ha seleccionado un margen más estricto para la calidad de audio debido a la mayor sensibilidad de los humanos a dicha componente. Este último punto en realidad incorpora la componente subjetiva de los usuarios que visualizan los contenidos y por tanto es susceptible de múltiples interpretaciones, esta es quizás la mayor ventaja de este tipo de sistemas de optimización: el permitir de manera sencilla especificar las preferencias de los usuarios de manera independiente para cada variable de respuesta. En este caso, y de acuerdo con las conclusiones obtenidas en [17] se ha optado por un modelo de utilidad combinado (correspondiente a los parámetros $S=2.0$) en el cual la componente de audio y de vídeo experimentan inicialmente una fuerte degradación (elevada pendiente) para ir paulatinamente moderando el nivel de degradación observado. Este comportamiento obedece al comportamiento de los usuarios que, en general, tienden a ser más tolerantes frente a sucesivas degradaciones de la calidad cuando dicha calidad se encuentra en niveles mínimos.

Para otro tipo de contenidos (deportes o películas) los usuarios valoran en mayor medida la calidad del factor visual, esto implicaría un modelo de utilidad más restrictivo en la

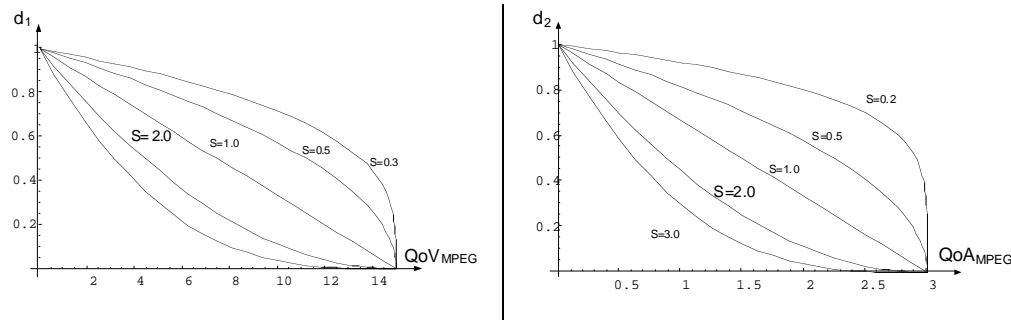


Fig. 3. Funciones de preferencia para las variables de respuesta QoE_{Video} y QoE_{Audio} . Tipo de contenido noticias

componente visual ($S=3.0$) frente a la auditiva ($S=2.0$), ver figuras 4.

Del mismo modo podría haberse contemplado la inclusión de otro tipo de restricciones adicionales como el tiempo de retardo en el cambio de secuencia, tiempo de retardo en el cambio de canal o funciones de coste para la calidad entregada. Finalmente la función de preferencia conjunta quedaría como sigue:

$$QoE = \sqrt{d_1(QoE_{Video}) \cdot d_2(QoE_{Audio})} \quad (3)$$

La cual, evaluada para la expresión dada en el apartado IV-B, proporciona los gráficos de contorno mostrados en la figura 5.

Llegados a este punto conviene destacar que la función de preferencia reflejada en 3 satisface de manera implícita las premisas del modelo de calidad presentado en [17]: se trata de un modelo multiplicativo, multimodal, dependiente del tipo de contenido y permite modelar dinámicamente el comportamiento de los usuarios frente a distintos niveles de degradación de calidad.

De acuerdo con las figuras, tanto el retardo como la pérdida de paquetes son los que más influyen en el nivel de calidad óptimo para la respuesta combinada de vídeo y audio para un ratio de codificación dado (áreas cuadradas). Hay que sealar que cuanto más alto es el ratio de codificación, más restrictivas se vuelven las condiciones.

A. Gestión de QoE en tiempo real

La métrica de calidad de vídeo y audio combinados -QoE- como se definió en la expresión 3 proporciona una expresión cuantitativa para desarrollar procedimientos de gestión de servicio basados en métricas de red observables para controlar la calidad proporcionada en servicios multimedia como vídeo bajo demanda. A continuación mediante un caso de estudio se muestra cómo una sesión multimedia establecida se gestiona en términos de calidad proporcionada.

Supongamos que un usuario está recibiendo una película codificada a 3500 Kilobits/s mediante H.264/AVC. Esto constituye un caso típico para un flujo que cumple el perfil ISMA de tipo-3. Según las métricas de red obtenidas para algún proceso de monitorización de red (bien sean pruebas de red, dispositivos de encaminamiento o informes RTCP -Real

Time Control Protocol-) las operaciones y el departamento de mantenimiento determinan que las condiciones de red actuales son $PLR = 0.06\%$ y Retardo = 30 ms. Como se puede ver en el gráfico de contornos central representado en la figura 5, el estado de la sesión real está en P_1 y por tanto el usuario está recibiendo un nivel de calidad cercano a las cero distorsiones.

Como resultado de la congestión de la red, debida al aumento de la demanda por algún evento especial, el grado de operación pasa abruptamente a presentar ratios de pérdidas de paquetes mayores y comienzan a producirse retardos que causan la degradación de la calidad, representada por P_2 . La primera medida posible que se puede tomar para restaurar el servicio a la situación anterior sería proporcionar recursos extra (tales como mecanismos de provisión de QoS, clústers de servidores, etc) para reducir tanto el retraso como la pérdida de paquetes. Otra alternativa factible sería imponer control de admisión limitando accesos de usuarios adicionales al servicio o incluso rechazando algunos usuarios existentes (por ejemplo abonados de niveles no garantizados). En nuestra opinión, estos dos mecanismos previos para afrontar la degradación de la calidad son en la práctica difíciles de implementar: el primero es técnicamente difícil de poner en práctica en tiempo real y el segundo es demasiado intrusivo.

Una tercera alternativa consiste en degradar el ratio de codificación de bits de las sesiones realizándolo en un punto específico del tiempo para restaurar la calidad a niveles óptimos, representado por P_3 . Esta alternativa permite restaurar el servicio sin ningún recurso extra y sin que los usuarios lo noten. Además, reduciendo la cantidad de información proporcionada ayuda a estabilizar la red y se necesita menos banda ancha para proporcionar el contenido. Hay que indicar, sin embargo, que para algún tipo de contenido como películas de alta definición o eventos deportivos, reducir el ratio de vídeo puede no ser aceptable para algunas audiencias sensibles. Este desarrollo es similar al propuesto en [20].

Otra forma de afrontar la degradación del servicio es mediante el denominado cambio de versión en la que una sesión multimedia establecida en cierto momento comienza a recibir contenidos que están codificados de otra forma. La reducción del ratio de bits puede considerarse un caso simple de cambio de versión. La selección del esquema de codificación más adecuado depende sobre todo de las características específicas

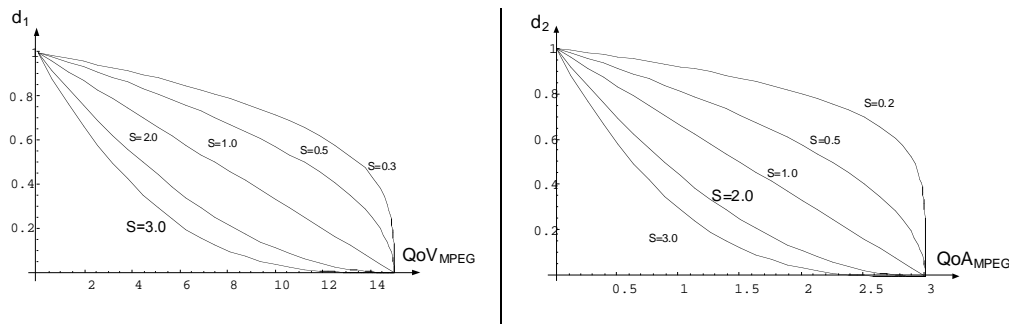


Fig. 4. Funciones de preferencia para las variables de respuesta QoE_{Video} y QoE_{Audio} . Tipo de contenido deportes

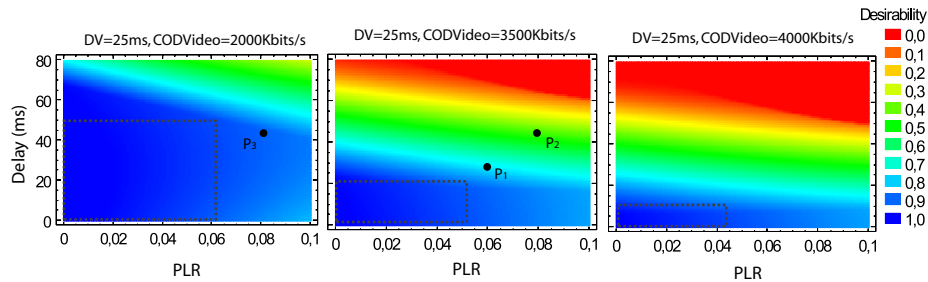


Fig. 5. Diagramas de contorno para la función de preferencia conjunta de H.264/AVC sobre RTP. Contenido Noticias

de la red y de si el proveedor del servicio tiene control para la provisión de mecanismos de QoS. El procedimiento de cambio de versión se puede realizar en tiempo real gracias a la existencia de SP y SI-Frames [34].

En el caso de redes subyacentes que proporcionen mecanismos de calidad de servicio (QoS) avanzados tales como Servicios Diferenciados [35] o cualquier forma de descarte de paquetes selectivo, sería posible codificar los contenidos multimedia en diversos subflujos conteniendo cada uno información cada vez más importante para el proceso de decodificación. Este esquema de codificación, denominado particionado de datos, permite una degradación del servicio gradual mediante descarte selectivo de los subflujos menos relevantes para el proceso de decodificación.

VI. CONCLUSIONES

Este trabajo define un método estadístico para caracterizar el comportamiento de los sistemas multimedia proporcionados sobre redes IP. Al mismo tiempo, incorpora, mediante funciones de utilidad, la percepción de los usuarios para permitir servicios multimedia orientados al usuario.

Nuestra propuesta define un procedimiento de adaptación en servicios multimedia capaz de asegurar la calidad entregada a los usuarios finales.

El modelo de utilidad contemplado es multimodal en el sentido de contemplar múltiples factores, fundamentalmente vídeo y audio, constituyentes de la percepción final de los usuarios

Debido a que el esquema propuesto se basa en métricas de red observables y controlables, es posible para los operadores de red monitorizar y controlar en tiempo real la

calidad proporcionada a sus abonados. Como líneas de trabajo futuras del desarrollo propuesto consideramos especialmente interesante profundizar en técnicas de control del servicio basadas en realimentación en el espacio de estados, de este modo sería posible calcular el impacto del número de accesos concurrentes al servicio o asignar de forma dinámica los recursos necesarios para la prestación del servicio.

AGRADECIMIENTOS

El presente trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia mediante el proyecto: CASERTEL-NGN (TSI2005-07306-C02-01).

REFERENCES

- [1] D. H. Yoon, B. Jeon: Indication of key picture in AVC compatible base layer Joint Video Team (JVT) of ISO/IEC MPEG, ITU-T VCEG (ISO/IEC JTC1/SC29/WG11, ITU-T SG16 Q.6 Input Document to JVT. (2005).
- [2] W. Li, F. Ling, X. Chen: Fine granularity scalability in MPEG-4 for streaming video. Circuits and Systems. (2000).
- [3] ISO/IEC 14496-10:2003, "Coding of Audiovisual Objects-Part 10: Advanced Video Coding". (2003).
- [4] T. Wiegand, H. Swartz, A. Joch, F. Kossentini: Rate-Constrained Coder Control and Comparison of Video Coding Standards. IEEE Transactions on Circuits and Systems for Video Technology, VOL. 13, No. 7. (2003).
- [5] S. Wenger: H.264/AVC over IP. IEEE Transactions on Circuits and Systems for Video Technology. (2003).
- [6] J. Ostermann, J. Bormans, P. List, et al.: Video coding with H.264/AVC: Tools, Performance, and Complexity. IEEE Circuits and Systems Magazine. (2004).
- [7] T. Stockhammer, M. M. Hannuksela, T. Wiegand: H.264/AVC in Wireless Environments. Transactions on circuits and systems for video technology. (2003).
- [8] O. Verscheure, P. Frossard, M. Hamdi: User-Oriented QoS Analysis in MPEG-2 Video Delivery. Journal of Real-Time Imaging (special issue on Real-Time Digital Video over Multimedia Networks), October 1999, vol.5, no 5, pp.305-314.

- [9] ITU-R BT.500: Methodology for the Subjective Assessment of the Quality of Television Pictures.
- [10] H. Knoche and D. McCarthy: Can Small be beautiful? Assessing Image Resolution Requirements for Mobile TV. *Journal ACM Multimedia*, 2005.
- [11] M.P. Hollier and A.N. Rimell: Multi-modal perception. *Journal of BT*, 1999.
- [12] J.G. Beerends and J. Stemerdink: A perceptual audio quality measure based on psychoacoustic sound representation. *Journal Audio Eng. Soc.*, 1992.
- [13] M.P. Hollier and M.O. Hawksford: Error activity and error entropy as a measure of psychoacoustic significance in the perceptual domain. *Proc. Inst. Elect. Vision, image and Signal Processing*. 1996.
- [14] S.A. Karunasekera and N.G. Kingsbury: A distortion measure for blocking artifacts in images based on human visual sensitivity. *IEEE Trans. image Processing*. 1995.
- [15] N.F. Dixon and L. Spitz: The detection of auditory visual desynchrony. *Journal of Perception*. 1999.
- [16] S. Rihs: The influence of Audio on Perceived Picture Quality and Subjective Audio-Video Delay tolerance. RACE-MOSAIC deliverable. 1995.
- [17] D.S. Hands: A basic multimedia quality model. *IEEE Transactions on multimedia*. 2004.
- [18] R. Venkatesh, Ajit S et al.: No reference metrics for video streaming applications. *Proceedings of the Packet Video Workshop*. (2004).
- [19] S. Kanumuri, P. Cosman, A. Reibman: A Generalized Linear Model for MPEG-2 Packet-Loss Visibility. *Proceedings of the Packet Video Workshop*. (2004).
- [20] G. Muntean, P. Perry, L. Murphy: Objective and Subjective Evaluation of QOAS Video Streaming over Broadband Networks. *IEEE Electronic Transactions on network and service management*. (2005).
- [21] TSG SA WG4 (Codec): Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs. 3GPP TS. (2006) 26.234.
- [22] A. Moorsel: Metrics for the Internet Age: Quality of Experience and Quality of Business. *Fifth Performability Workshop*. (2001).
- [23] W. Walsh, G. Tesauro, J. O. Kephart, et al.: Utility functions in Autonomic Systems. *Proceedings of the International Conference on Autonomic Computing*. (2004).
- [24] ISO/IEC 20000-1:2005: Information technology, Service management, Part 1: Specification. ISO published standard. (2005).
- [25] K. Yang, B. El-Haik: *Design for Six Sigma* McGraw-Hill. ISBN 0071412085. (2003).
- [26] National Institute of Standards and Technology (NIST). e-Handbook of Statistical Methods. [Online]. <http://www.itl.nist.gov/div898/handbook/> (2005).
- [27] Network emulation (NETEM). [Online]. Available at: <http://linux-net.osdl.org/index.php/Netem>
- [28] Darwin Streaming Server. [Online]. Available at: <http://developer.apple.com/opensource/server/streaming/index.html>
- [29] ITU-R BT Quality of service ranking and measurement methods for digital video broadcasting services delivered over broadband internet protocol networks. ITU-T Recommendation, bt.6/bl/27 (07/05)
- [30] ISMA Technical Committee: Internet Streaming Media Alliance Implementation Specification, Version 1.0. Implementation Specification. (2001)
- [31] ISMA Technical Committee: Internet Streaming Media Alliance Implementation Specification, Version 2.0. Implementation Specification. (2005).
- [32] Derringer G., Suich R. Simultaneous Optimization of Several Response Variables. *Journal of Quality Technology*, **12** 4 (1980).
- [33] T. Wiegand, G. Sullivan, G. Bjntegaard, A. Luthra: Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 13, No. 7. (2003).
- [34] M. Karczewicz, R. Kurceren: The SP- and SI-Frames Design for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 13, No. 7. (2003).
- [35] IETF Working Group: Differentiated Services (diffserv). [online]. <http://www.ietf.org/html.charters/OLD/diffserv-charter.html>

MODELADO DINÁMICO DE PLANTAS SOLARES DE DESALACIÓN: PROYECTO AQUASOL

L.Roca*

L. Yebra*, M. Berenguel†, D. Alarcón*

*Plataforma Solar de Almería (PSA), Carretera Senés s/n, 04200 Tabernas (Almería)

†Universidad de Almería. Dpto. de Lenguajes y Computación. Ctra. Sacramento s/n, 04120 Almería
lidia.roca@psa.es, luis.yebra@psa.es, beren@ual.es, diego.alarcon@psa.es

Resumen

El proyecto AQUASOL constituye en la actualidad uno de los objetivos más ambiciosos dentro de la tecnología de desalación de agua de mar, utilizando como fuente de energía la radiación solar. Dicho proyecto, financiado por la Unión Europea, coordinado por CIEMAT y en desarrollo en la Plataforma Solar de Almería, estudia el proceso de producción de agua destilada mediante energía solar, con el objetivo fundamental de minimizar los costes y el uso de energías fósiles de apoyo. Como toda planta de producción, es necesario conocer el comportamiento de los distintos subsistemas que la forman para desarrollar técnicas de control apropiadas y que mejoren la eficiencia energética global del proceso de desalación. Uno de los objetivos fundamentales ha sido el desarrollo de modelos dinámicos derivados de primeros principios, de cada uno de los subsistemas, su posterior calibración e interconexión para formar el modelo completo. En este trabajo se presentan los resultados preliminares, en materia de modelado dinámico y calibración, de cada uno de los subsistemas.

Palabras clave: modelado, control, energía solar

1. INTRODUCCIÓN

El proyecto AQUASOL consta de una planta de desalación del tipo multi efecto acoplada a un campo de colectores tipo CPC [1]. Mediante el uso de la energía solar se desea obtener una temperatura del agua de alimentación adecuada y, de esta forma asegurar la producción de agua desalada. Como todo proceso de producción con un objetivo comercial, es importante minimizar los costes asegurando un nivel de producto óptimo. Es en este ámbito donde se hace básico el uso de técnicas de control adecuadas que respeten las restricciones que cada sistema muestra. Para ello, es importante caracterizar cada uno de los subsistemas de los que está compuesta la planta y modelar la dinámica que en ellos se presenta.

En la primera sección de este trabajo se muestra una visión general de la motivación que conlleva el uso de técnicas de control en el proyecto

AQUASOL. Con el objetivo de desarrollar dichos controladores, se explican los modelos básicos desarrollados en el siguiente apartado. Por último se muestran los resultados obtenidos de los modelos en distintos días de ensayo.

2. MOTIVACIÓN DEL USO DE TÉCNICAS DE CONTROL

Con el objetivo de poner en funcionamiento la planta, es necesario definir los lazos de control básicos de operación así como dispositivos de emergencia que impidan el paso por puntos de operación con riesgos. Una vez que el sistema se encuentra en disposición de poder trabajar manualmente bajo la supervisión de operadores, pueden diseñarse lazos de control que optimicen la producción.

Uno de los posibles riesgos que existe en la planta, es las temperaturas extremas que sufren los colectores. En concreto, es importante que la temperatura en el campo no baje demasiado durante la noche y evitar así la congelación de los colectores. Actualmente hay implementado un dispositivo de emergencia mediante una máquina de estados que hace poner en funcionamiento la bomba del campo solar cuando se detecta una temperatura inferior a la mínima definida. Dicha bomba extrae el agua de los tanques de almacenamiento que están a temperaturas superiores, y actúa durante un tiempo determinado hasta que se completa la renovación del agua en la totalidad del campo. A simple vista este lazo no muestra ninguna complejidad mayor, pero es necesario hacer un estudio de la dinámica del campo con el fin de encontrar la relación caudal-tiempo de funcionamiento de la bomba y así reducir el consumo energético de la misma. Un caso similar al anterior es el riesgo de evaporación del agua en el campo por altas temperaturas que obligan a circular caudal con el objetivo de bajar la temperatura, pero intentando reducir los costes energéticos.

Por otra parte, según los fabricantes de los colectores, para obtener el máximo rendimiento del campo solar tiene que existir un incremento de temperatura entre la entrada y la salida del campo de 10°C. Según sus especificaciones se diseño una

máquina de estados de forma que si la temperatura del campo supera en más de 8°C la de entrada, la bomba del campo se para y volverá a arrancarse cuando la diferencia sea menor de 3°C . Con este mecanismo no es posible trabajar continuamente en el punto de operación óptimo, sino que éste se alcanza a intervalos. Es por ello por lo que un modelo del campo solar se hace indispensable para poder desarrollar técnicas de control que aseguren la operación sobre el punto de trabajo definido. Por último, uno de los factores que afecta a la producción de agua destilada es la temperatura de entrada al primer efecto. Por ello es importante obtener la dependencia de esta variable en función de las perturbaciones relacionadas.

3. MODELOS DE LA PLANTA AQUASOL

La planta AQUASOL (figura 1) está formada por una planta de desaladora multi efecto (MED) [5] que recibe agua de alimentación procedente de un tanque de agua primario. Dicho tanque está conectado en serie a un tanque secundario del que fluye agua hacia el campo solar. Tras elevarse la temperatura del agua mediante energía solar, el fluido retornará al tanque primario o secundario dependiendo de la temperatura que se desee alcanzar en el tanque primario. Existe además una bomba de absorción de doble efecto (BADE) [5] para asegurar la continua operación de la planta. En esta sección se muestra brevemente el modelo desarrollado en el campo solar para obtener la temperatura de salida, el modelo de los tanques de almacenamiento para estimar la temperatura de entrada al primer efecto de la planta MED, y el modelo del intercambiador del primer efecto para conocer la temperatura de salida del primer efecto.

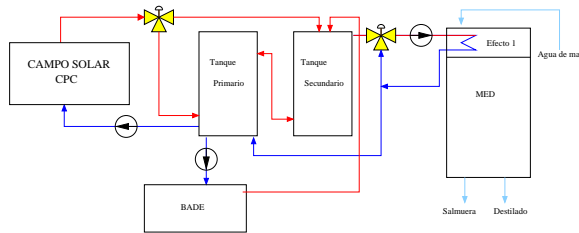


Figura 1: Esquema de la planta AQUASOL

3.1. MODELO DEL CAMPO SOLAR

Con el objetivo de alcanzar equilibrio hidráulico en el campo de colectores, los 252 CPCs están divididos en cuatro filas de 63 colectores. Dentro de cada fila, los colectores están comunicados en paralelo por grupos de nueve, que a su vez tienen

conexiones en serie y paralelo por grupos de tres como se muestra en la figura 2.

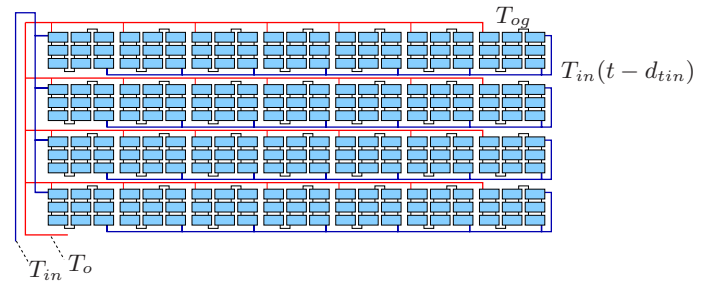


Figura 2: Campo Solar

En el modelo del campo solar se considera que un grupo de nueve colectores es equivalente a un hipotético tubo con una longitud, L_{eq} , tres veces la de los tubos reales que los colectores poseen L_o , $L_{eq} = 3L_o$. Además, el flujo másico de entrada a dicho grupo, \dot{m}_{eq} , es una fracción del flujo másico total que entra al campo \dot{m}_F , $\dot{m}_{eq} = \dot{m}_F \cdot (4 \cdot 7 \cdot 7 \cdot 3)^{-1}$.

El comportamiento dinámico de dicho tubo equivalente de colector puede ser determinado mediante un modelo no lineal de parámetros distribuidos basado en ecuaciones diferenciales.

$$\rho \cdot C_p \cdot A \cdot \frac{\partial T_{og}}{\partial t} = I \cdot \eta \cdot C \cdot G - H(\bar{T} - T_a) - \rho \cdot C_p \cdot \dot{m}_{eq}(t) \cdot \frac{T_{og} - T_{in}(t - d_{tin})}{L} \quad (1)$$

$$\bar{T} = \frac{T_{og} + T_{in}(t - d_{tin})}{2} \quad (2)$$

donde T_{og} es la temperatura de salida en un grupo de nueve colectores y T_{in} la temperatura de entrada al campo. Además estas ecuaciones tienen en cuenta características geométricas del tubo de colector (apertura G , sección transversal A , longitud L), propiedades del metal (concentración C) y del fluido (densidad ρ , capacidad calorífica C_p), radiación I y coeficiente global de transferencia de calor H . Es importante señalar que entre el sensor de temperatura en la entrada del campo solar y la entrada a un grupo de nueve colectores hay un retardo temporal, d_{tin} , que varía dependiendo de la velocidad del fluido. Es posible modelar este retraso utilizando la distancia l_1 que el fluido tiene que recorrer para llegar a la entrada del tubo de colector [3]:

$$l_1 = \int_0^{d_{tin}} v(t) dt \quad (3)$$

Discretizando la ecuación en tiempos de muestreo,

$$l_1 = \frac{T_s}{S} \sum f(i) \quad (4)$$

donde f es el caudal del agua en la entrada del campo de colectores. De esta forma, conociendo l_1 es posible calcular el retardo existente entre los dos puntos debido a la velocidad del fluido en los instantes anteriores.

Para obtener la temperatura de salida del campo de colectores (T_o) debe ser considerado otro retardo en la temperatura de salida del grupo de nueve colectores (T_{og}) que depende del flujo másico al igual que ocurría con la temperatura de entrada.

$$T_o = T_{og}(t - d_{to}) \quad (5)$$

El término de las pérdidas térmicas globales puede ser representado por un polinomio de segundo grado con $(T_m - T_a)$ como variable independiente [2]. Con el objetivo de simplificar el modelo, este término se ha asumido constante con un valor obtenido experimentalmente en distintos días de operación.

3.2. TANQUES DE ALMACENAMIENTO

Mediante balances de masa y energía es posible obtener un modelo en espacio de estados de cuarto orden. Para ello se supone despreciable los efectos de estratificación, pudiendo por tanto definir una única temperatura en cada tanque. Además, las pérdidas térmicas se supondrán despreciable como primera aproximación. Las variables de estado son los niveles y temperaturas en los dos tanques (L_1, L_2, T_1, T_2), y como salida del modelo se obtiene la temperatura de entrada al primer efecto de la planta MED (T_{iM}). Por tanto, el modelo de los tanques es representado mediante dos funciones F y G que dependen no linealmente de las variables de estado \mathbf{x} y las entradas \mathbf{u} :

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u}) \quad (6)$$

$$\mathbf{y} = G(\mathbf{x}, \mathbf{u}) \quad (7)$$

donde $\mathbf{u} = (\dot{m}_F, \dot{m}_{ref}, \dot{m}_D, \dot{m}_{21}, \gamma, T_{oM}, T_D)$.

$$\dot{L}_1 = \frac{1}{\rho A} [-\dot{m}_F + \dot{m}_{21} + \gamma \dot{m}_{ref} - \dot{m}_D] \quad (8)$$

$$\dot{L}_2 = \frac{1}{\rho A} [\dot{m}_F - \dot{m}_{21} - \gamma \dot{m}_{ref} + \dot{m}_D] \quad (9)$$

$$\dot{T}_1 = \frac{1}{\rho A L_1} [-T_1(\dot{m}_{21} + \gamma \dot{m}_{ref}) + \dot{m}_{21} T_{12} + \gamma \dot{m}_{ref} T_{oM}] \quad (10)$$

$$\dot{T}_2 = \frac{1}{\rho A L_2} [-T_2(\dot{m}_F + \dot{m}_D - \dot{m}_{21}) + \dot{m}_F T_F + \dot{m}_D T_D - \dot{m}_{21} T_{12}] \quad (11)$$

El agua de entrada es una mezcla entre el agua procedente del tanque primario y el retorno de la propia celda (figura 3).

$$T_{iM} = \gamma T_2 + (1 - \gamma) T_{oM} \quad (12)$$

Estas ecuaciones tienen en cuenta los flujos másicos en el campo solar, MED y BADE ($\dot{m}_F, \dot{m}_{ref}, \dot{m}_D$), así como el flujo másico entre los dos tanques \dot{m}_{21} . El parámetro γ se encuentra en el rango $[0,1]$ y es utilizado para modelar la regulación de la válvula. Dependiendo de las condiciones de operación, la variable T_{12} puede variar entre T_1 y T_2 en cada iteración. Por último T_{oM} es la temperatura de salida del primer efecto y que en la sección siguiente se modela. Hay que tener en cuenta que el flujo entre los tanques \dot{m}_{21} es una variable algebraica que depende de los niveles de los tanques y propiedades termodinámicas del fluido [4].

3.3. PRIMER EFECTO DE LA PLANTA MED

La planta MED está conectado a un sistema de inyectores que hace que en la primera celda se trabaje en condiciones de saturación. Por tanto, la temperatura en la primera celda (T_{cell1}) es la temperatura de saturación. Aplicando un balance de calor en estacionario es posible estimar la temperatura de salida del agua en la primera celda T_{oM} .

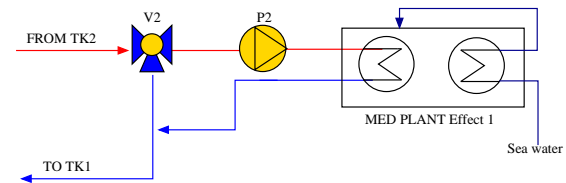


Figura 3: Primer efecto de la planta MED

$$\dot{m}_{ref} \cdot c_p \cdot (T_{iM} - T_{oM}) = k \cdot A (T_m - T_{cell1}) \quad (13)$$

$$T_m = \frac{T_{iM} + T_{oM}}{2} \quad (14)$$

donde el parámetro k es el coeficiente de transferencia de calor global. Hay que señalar que el producto $k \cdot A$ ha sido evaluado experimentalmente, y que el modelo supone que la temperatura del metal (T_m) es la media entre los valores de la temperatura del agua a la entrada (T_{iM}) y salida (T_{oM}) del primer efecto.

4. RESULTADOS

Los resultados de los modelos son mostrados para el caso de distintos días de operación trabajando únicamente con el campo solar como medio de aporte energético.

4.1. MODELO DEL CAMPO SOLAR

En la figura 4 se observan las perturbaciones para el día 06/03/06 en el que, aunque la radiación no presenta variaciones relevantes, se actuó sobre la bomba para conseguir diferentes escalones en caudal. Además, la temperatura de entrada del campo presenta un escalón que corresponde con la puesta en funcionamiento de la planta MED.

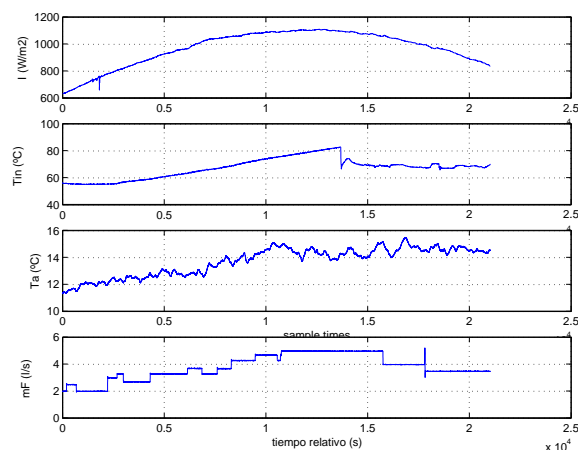


Figura 4: Perturbaciones

Es posible comparar la temperatura que el modelo predice con los valores reales mediante la gráfica 5. Se observa que ambas curvas siguen una misma dinámica en función de los cambios en las perturbaciones y no presentan un retardo importante.

4.2. MODELO DE LOS TANQUES DE ALMACENAMIENTO

En el caso de los tanques de almacenamiento, las perturbaciones que se muestran en la figura 6 son el caudal en el campo solar, regulación de la válvula de mezcla γ y el caudal de entrada y temperatura de salida en primer efecto de la MED.

En los instantes iniciales, figura 7, el modelo no se ajusta a los valores reales debido a que la planta

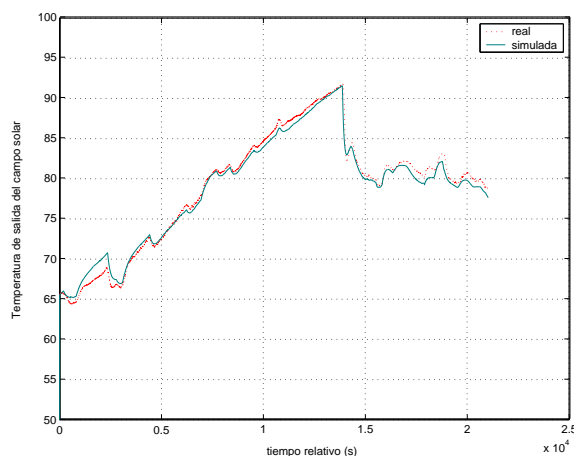


Figura 5: Temperatura de salida del campo solar

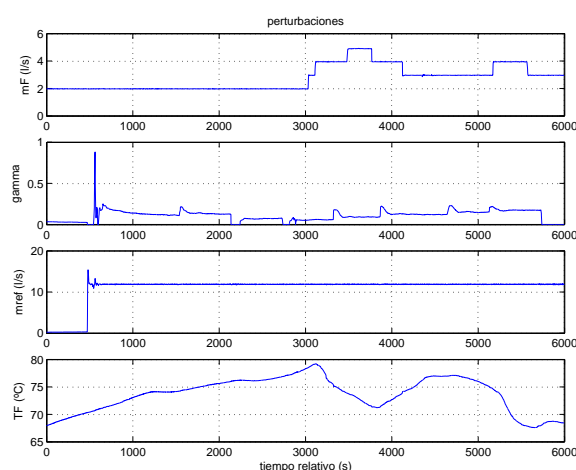


Figura 6: Perturbaciones

MED no está en funcionamiento. En esos instantes el caudal que circla es nulo y, por tanto, la temperatura de entrada al primer efecto es la de un fluido estancado y sometido a pérdidas con el ambiente.

En el instante en el que empieza a circular caudal a través del primer efecto, la temperatura sube en función de la regulación de la válvula y la temperatura de salida.

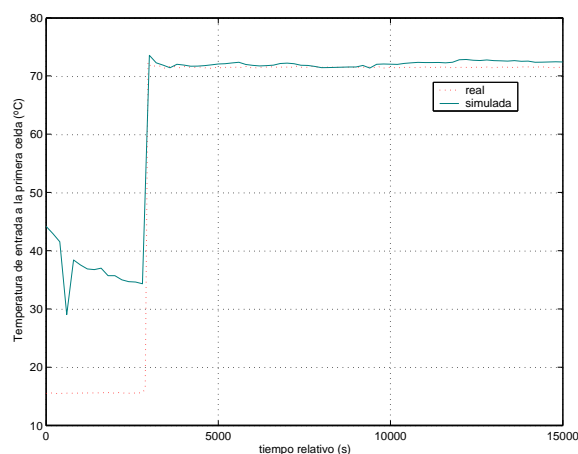


Figura 7: Temperatura de entrada al primer efecto de la planta MED

4.3. MODELO DEL PRIMER EFECTO DE LA PLANTA MED

Las perturbaciones mostradas en la figura 8 corresponden al día 23/06/06 en el que se muestran variaciones en forma de escalón en la temperatura de entrada al primer efecto. Es por ello por lo que la temperatura de salida (figura 9) tiene también una dinámica de escalones. Las diferencias que se observan en las curva del modelo respecto a la real, se localizan al producirse los cambios en la temperatura de entrada. Estas discrepancias son debidas al controlador PI que regula dicha temperatura de entrada actuando sobre la válvula reguladora, y tienen su origen en la simplificación realizada a la hora de escoger un modelo estacionario y no dinámico.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo presenta el estado actual de las tareas de desarrollo de modelados dinámicos de la planta de desalación basada en energía solar del proyecto AQUASOL, desarrollado en la Plataforma Solar de Almería. El objetivo de estos modelos es el de servir como base de controladores avanzados que permitan optimizar la eficiencia de dicha planta en diferentes condiciones de operación.

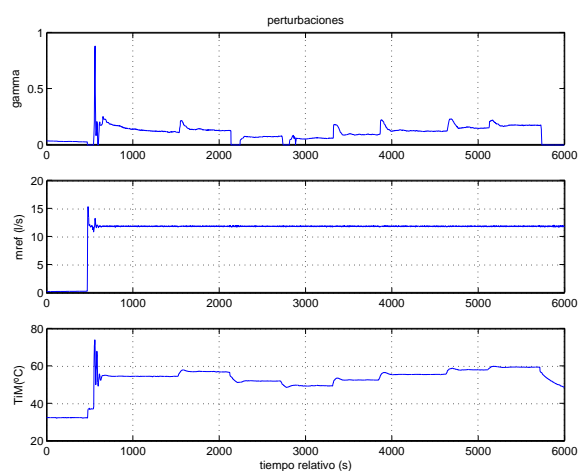


Figura 8: Perturbaciones

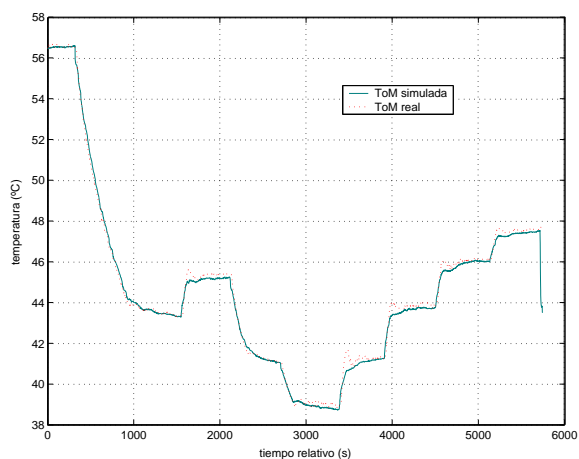


Figura 9: Temperatura de salida del primer efecto de la planta MED

Se han descrito los submodelos para el campo de colectores, tanques de almacenamiento y primer efecto de la planta MED que se han estudiado hasta el momento para una vez finalizada la validación, englobarlos en un único modelo que simule la planta en su totalidad.

Agradecimientos

Los autores de este trabajo desean dar las gracias a la Comisión Europea por el soporte financiero para el proyecto AQUASOL (Contrato no. EVK1-CT2001-00102), a los fondos CICYT-FEDER (proyectos DPI2002-04375-C03 y DPI2004-07444-C04-04) y a la Junta de Andalucía Consejería de Innovación, Ciencia y Empresa. Este trabajo también se ha realizado en el ámbito del acuerdo específico de colaboración entre la Plataforma Solar de Almería y el Grupo de Investigación *Automática, Electrónica y Robótica* de la Universidad de Almería (TEP197) titulado *Desarrollo de sistemas y herramientas de control para plantas termosolares*.

Referencias

- [1] Duffie, J.A, (1991) Solar Engineering of Thermal Process, A Wiley Interscience Publication.
- [2] Fraindenraich, N., (1999) "Simulation model of a CPC collector with temperature-dependent heat loss coefficient", *Solar Energy*, pp 99-110.
- [3] Normey, J.E, (1998) "A Robust adaptive dead-time compensator with application to a solar collector field" IFAC International Workshop on Linear Time Delay Systems.
- [4] Roca, L., (2005) "Obtención de modelos para plantas desaladoras basadas en energía solar" Jornadas de Automática.
- [5] Zarza, E, (1995) Solar Thermal Desalination Project Phase II Results & Final Project Report. Colección Documentos Ciemat.

CONTROL VISUAL BASADO EN IMAGEN: UN NUEVO MÉTODO PARA ESTIMAR EL JACOBIANO DE LA IMAGEN EN ENTORNOS DINÁMICOS

L. Pari¹, C. González¹, J.M. Sebastián¹, L. Angel² y V. Mut³

¹Departamento de Automática, Ingeniería Electrónica e Informática Industrial (DISAM)
Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid
C/ José Gutiérrez Abascal, 2, 28006 Madrid, España

²Facultad de Ingeniería Electrónica, Universidad Pontificia Bolivariana
Km. 7 Vía de Piedecuesta, Bucaramanga, Colombia

³Instituto de Automática, Universidad Nacional de San Juan
Av. San Martín Oeste 1109, 5400 San Juan, Argentina

lpari@etsii.upm.es, cgpascual@etsii.upm.es, langel@etsii.upm.es, jsebas@etsii.upm.es,
vmut@inaut.unsj.edu.ar

Resumen

El presente artículo describe varios novedosos algoritmos para la estimación on-line del Jacobiano de la imagen. Se ha aplicado exitosamente para tareas de seguimiento de un sistema articular no calibrado y con tres grados de libertad, empleando dos cámaras fijas en el espacio y débilmente calibradas. Los algoritmos propuestos se han mostrado especialmente robustos cuando las características de la imagen se calculan con un nivel de ruido medio, y los resultados obtenidos superan claramente a los hallados con los algoritmos existentes en la literatura especializada.

Palabras Clave: Jacobiano de la Imagen, control visual dinámico, sistema no calibrado, matriz fundamental.

1 INTRODUCCIÓN

La realización de tareas por parte de los sistemas robotizados en entornos estructurados con presencia de objetos cuya posición y orientación es perfectamente conocida, es un problema suficientemente estudiado en la actualidad [9]. Sin embargo, la realización de tareas en entornos no estructurados o dinámicos presenta numerosas dificultades aún no suficientemente resueltas. Un sistema sensorial de visión es capaz de aportar una información extremadamente útil en estos entornos puesto que ofrece información acerca de cuáles son los objetos presentes en la escena de trabajo y además y quizás más importante, permiten determinar de una forma suficientemente precisa su posición, orientación, y velocidad.

Con el nombre de control visual se agrupan todas las técnicas que utilizan la información proporcionada

por sensores visuales (por ejemplo, cámaras) en el control de robot o sistemas articulares. En los últimos años ha sido objeto de múltiples estudios, existiendo en la literatura especializada interesantes surveys como los descritos en [2], [7] y [8]. La principal clasificación entre los sistemas que emplean control visual radica en el propio esquema de control. Así existe el Control Visual Basado en Posición (PBVS) donde la señal de error y la ley de control se expresan en referencias cartesianas en función de la posición y orientación deseadas y medidas. También se le conoce como 3D visual servoing. Por el contrario en el Control Visual Basado en Características (IBVS) la señal de error y la ley de control se expresan en el plano de la imagen en función de las características visuales deseadas y medidas. Se le conoce en la literatura como 2D visual servoing. Implica el cálculo del llamado Jacobiano de la imagen, que relaciona linealmente la variación de las características visuales con la variación del sistema articular. Dicho Jacobiano puede ser calculado analíticamente o se puede obtener mediante una estimación. El PBVS posee como ventajas un control desacoplado, la ausencia de singularidades, y un mejor control de la trayectoria de la cámara, imprescindible en entornos complejos. El IBVS presenta como ventajas un conocimiento menor del modelo tridimensional del entorno, una menor sensibilidad al ruido y a los errores de calibración y una mejor precisión.

Este artículo presenta un método para la estimación del Jacobiano on-line con dos cámaras fijas observando la escena, sin necesidad de requerir la calibración euclídea de las cámaras ni la calibración cinemática del sistema articular. Su principal novedad radica en incorporar la información de la llamada calibración débil o calibración proyectiva obtenida mediante el cálculo de la matriz fundamental, lo que permite una estimación más robusta ante el ruido presente en la detección de las características visua-

les. La tarea propuesta en las pruebas realizadas es conseguir que las características visuales, centroides de la proyección de unos puntos, cumplan unas especificaciones impuestas. Más detalles están expuestos en González [5] y Sebastián [13].

El artículo se estructura en los siguientes apartados: Tras la presente introducción, en el apartado segundo se detallan la nomenclatura y los conceptos teóricos utilizados en el artículo. En el apartado tercero se exponen los algoritmos novedosos propuestos, mientras que en el apartado cuarto se describe el entorno utilizado en las pruebas y los resultados obtenidos, para por último en el apartado quinto se destacan las conclusiones obtenidas.

2 JACOBIANO DE LA IMAGEN

Supóngase que se observa un sistema articular o robot desde una o varias vista fijas. Sea $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_p]^T$ el vector de dimensión p que representa la posición del efector final en un sistema de coordenadas cartesiano. Sea $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$ el vector de dimensión n que representa la posición articular del robot. Sea $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_m]^T$ el vector de dimensión m que representa las características de las imágenes (por ejemplo las coordenadas de un punto en una o varias imágenes).

La relación entre la velocidad articular del robot $\dot{\mathbf{q}} = [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_n]^T$ y su correspondiente velocidad en el espacio de la tarea $\dot{\mathbf{r}} = [\dot{r}_1 \ \dot{r}_2 \ \dots \ \dot{r}_p]^T$, se expresa en términos del Jacobiano del robot \mathbf{J}_{rq} , como $\dot{\mathbf{r}} = \mathbf{J}_{rq} \dot{\mathbf{q}}$. La relación entre la velocidad de las características $\dot{\mathbf{s}} = [\dot{s}_1 \ \dot{s}_2 \ \dots \ \dot{s}_m]^T$ y la velocidad en el espacio de la tarea se expresa como $\dot{\mathbf{s}} = \mathbf{J}_{sr} \dot{\mathbf{r}}$.

La velocidad de las características de la imagen se puede relacionar directamente con las velocidades articulares en términos de un Jacobiano compuesto, conocido como Jacobiano visual-motor ([3] y [14]):

$$\dot{\mathbf{s}} = \mathbf{J}_{sq} \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial s_1}{\partial q_1} & \dots & \frac{\partial s_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_m}{\partial q_1} & \dots & \frac{\partial s_m}{\partial q_n} \end{bmatrix} \dot{\mathbf{q}} \text{ con } \mathbf{J}_{sq} = \mathbf{J}_{sr} \mathbf{J}_{rq} = \mathbf{J} \quad (1)$$

La determinación analítica de esta matriz no es sencilla. Es necesario destacar que en su cálculo intervienen: los parámetros intrínsecos de la calibración de la

cámara (distancia focal, coordenadas centrales de la imagen), la reconstrucción tridimensional del punto o una aproximación (coordenada Z), la calibración cinemática de la cámara (relación entre las coordenadas de la cámara y el origen articular), y la calibración cinemática del robot. La mayoría de los trabajos previos en control visual suponen conocidos la estructura y los parámetros del sistema, o que los parámetros pueden identificarse en un proceso off-line. Un esquema de control con identificación de parámetros off-line no es robusto ante perturbaciones, cambio de parámetros, y entornos desconocidos. El control visual basado en imagen sin calibración consiste en la estimación dinámica del Jacobiano visual-motor durante el movimiento.

2.1 ESTIMACIÓN DEL JACOBIANO

La literatura especializada recoge cuatro métodos para la estimación del Jacobiano descrito en (1) ([3] y [14]). En todos los casos se parte de un Jacobiano inicial obtenido al realizar n pequeños movimientos linealmente independientes.

2.1.1 Estimación basada en los últimos movimientos

Si se representa respectivamente el incremento de características de la imagen y del movimiento articular por $\Delta \mathbf{s}_k = \mathbf{s}_k - \mathbf{s}_{k-1}$ y por $\Delta \mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}$ y se supone que el Jacobiano de la imagen es constante durante n movimientos, entonces el Jacobiano puede definirse como la matriz que satisface simultáneamente n movimientos:

$$\begin{bmatrix} \Delta \mathbf{s}_{k-n+1} \\ \vdots \\ \Delta \mathbf{s}_k \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{q}_{k-n+1} \\ \vdots \\ \Delta \mathbf{q}_k \end{bmatrix} \mathbf{J}_k^T \quad (2)$$

Este método puede presentar problemas si los movimientos articulares se producen en la misma dirección, algo que puede ser bastante normal. En estos casos se obtiene una matriz mal condicionada, con todos los problemas que ello conlleva. Zutanito en [14] resuelve el problema detectando cuándo se produce esta situación y añadiendo pequeños movimientos exploratorios.

2.1.2 Método de Broyden

En este método el Jacobiano se estima de forma recursiva, a partir del anterior Jacobiano, añadiendo la información del último movimiento. Con respecto al anterior método tiene la ventaja de aglutinar la información de todos los movimientos. La ecuación para el método de Broyden ([11] y [1]) es:

$$\mathbf{J}_k = \mathbf{J}_{k-1} + \frac{\left(-\Delta \mathbf{e}_k + \frac{\partial \mathbf{e}_k}{\partial t} \Delta t - \mathbf{J}_{k-1} \Delta \mathbf{q}_k \right) \Delta \mathbf{q}_k^T}{\Delta \mathbf{q}_k^T \Delta \mathbf{q}_k} \quad (3)$$

siendo $\mathbf{e}_k = \mathbf{s}_k^* - \mathbf{s}_k$ el error en las características de la imagen, y \mathbf{s}_k^* las características deseadas.

2.1.3 Mínimos cuadrados recursivos (RLS)

En este método el Jacobiano es estimado recursivamente por medio del algoritmo de mínimos cuadrados ([11] y [1]). La fórmula que define el Jacobiano es:

$$\mathbf{J}_k = \mathbf{J}_{k-1} + \frac{\left(-\Delta \mathbf{e}_k + \frac{\partial \mathbf{e}_k}{\partial t} \Delta t - \mathbf{J}_{k-1} \Delta \mathbf{q}_k \right) \Delta \mathbf{q}_k^T \mathbf{P}_{k-1}}{\lambda + \Delta \mathbf{q}_k^T \mathbf{P}_{k-1} \Delta \mathbf{q}_k} \quad (4)$$

Donde

$$\mathbf{P}_k = \frac{1}{\lambda} \left(\mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1} \Delta \mathbf{q}_k \Delta \mathbf{q}_k^T \mathbf{P}_{k-1}}{\lambda + \Delta \mathbf{q}_k^T \mathbf{P}_{k-1} \Delta \mathbf{q}_k} \right) \quad (5)$$

Las prestaciones del método dependen del parámetro λ , que varía en el rango entre 0 y 1, y pondera los movimientos previos. Permite un compromiso entre la información suministrada por todos los antiguos movimientos y los nuevos datos, posiblemente corruptos por el ruido. En presencia de ruido moderado, valores de λ cercanos a 0.9 son usados normalmente.

2.1.4 Filtro de Kalman

Otra forma de estimar recursivamente el Jacobiano es utilizando el filtro de Kalman ([12]). El sistema es modelado utilizando variables de estado como:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\eta}_k \\ \mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \nu_k \end{cases} \quad (6)$$

Donde $\boldsymbol{\eta}_k, \nu_k$ son respectivamente el ruido del estado y de la medida, \mathbf{x} es un $(mn \times 1)$ vector formado por la concatenación de los elementos de las filas del Jacobiano:

$$\mathbf{x} = \begin{bmatrix} \frac{\partial s_1}{\partial q} & \frac{\partial s_2}{\partial q} & \dots & \frac{\partial s_m}{\partial q} \end{bmatrix}^T \quad (7)$$

Donde $\frac{\partial s_i}{\partial q}$ es la i -th fila del Jacobiano \mathbf{J}_k . El vector de medida es:

$$\mathbf{y}_{k+1} = \mathbf{s}_{k+1} - \mathbf{s}_k \quad (8)$$

Siendo la matriz \mathbf{C}_k el incremento articular.

$$\mathbf{C}_k = \begin{bmatrix} \Delta \mathbf{q}_k^T & & 0 \\ & \ddots & \\ 0 & & \Delta \mathbf{q}_k^T \end{bmatrix} \quad (9)$$

Las ecuaciones del filtro de Kalman son:

$$\begin{aligned} \mathbf{P}_{k+1}^- &= \mathbf{P}_k + \mathbf{R}_\eta \\ \mathbf{K}_{k+1} &= \mathbf{P}_{k+1}^- \mathbf{C}_k^T [\mathbf{C}_k \mathbf{P}_{k+1}^- \mathbf{C}_k^T + \mathbf{R}_\nu]^{-1} \\ \mathbf{P}_{k+1} &= [\mathbf{I} - \mathbf{K}_{k+1} \mathbf{C}_k] \mathbf{P}_{k+1}^- \\ \hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{x}}_k + \mathbf{K}_{k+1} [\mathbf{y}_{k+1} - \mathbf{C}_k \hat{\mathbf{x}}_k] \end{aligned} \quad (10)$$

Donde \mathbf{P}_k es la matriz de covarianza en el momento k , \mathbf{P}_{k+1}^- es la predicción de \mathbf{P}_k , $\mathbf{R}_\eta, \mathbf{R}_\nu$ son las matrices de covarianza del estado y de la medida respectivamente y \mathbf{K}_{k+1} es la ganancia de Kalman.

2.2 JACOBIANO DE MÚLTIPLES VISTAS

Cuando se utilizan varias vistas, el visual-motor Jacobiano se puede definir como la concatenación de cada vista (más detalle en [10] y [1]). Todos los Jacobianos poseen los mismos incrementos articulares, aunque distintas características. Si las características son las proyecciones de puntos, es posible incorporar la restricción epipolar en la estimación del Jacobiano, como se muestra en el presente artículo. La notación usada se describe con más detalle en la sección 3.2.

En los trabajos previos ([13]) se comparan los resultados obtenidos al estimar el Jacobiano usando una o dos cámaras. Los resultados muestran una importante mejora en el segundo caso, para cualquier método. En la mayoría de las aplicaciones las ventajas superan los posibles efectos negativos de las desventajas: incremento en el coste de los equipos y en el

gasto computacional. Los experimentos descritos en el presente trabajo, se han realizado utilizando dos cámaras.

2.3 LEY DE CONTROL

La ley de control utilizada añade al termino proporcional ([11]) un término predictivo a partir de las dos últimas referencias según la ecuación:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^+ (\mathbf{s}_k^* - \mathbf{s}_k + \mathbf{s}_k^* - \mathbf{s}_{k-1}^*) \quad (11)$$

Donde $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ es la pseudo-inversa de la Jacobiana de la imagen.

3 ALGORITMOS PROPUESTOS

La línea de investigación seguida en el artículo es la estimación del Jacobiano basándose en los movimientos ya efectuados, no necesariamente los n últimos realizados. Se emplea la información visual suministrada por dos cámaras. Dos son las novedades aportadas: por un lado se ha dotado a cada movimiento de una fiabilidad, a fin de poder elegir entre los movimientos ya efectuados los más adecuados o fiables. Por otro se ha incorporado al cálculo del Jacobiano la restricción epipolar que debe de cumplir la proyección de un mismo punto en dos imágenes (apartado 3.2), lo que aumenta significativamente la robustez del método, como se verá en el cuarto apartado.

3.1 ESTIMACIÓN DE LA FIABILIDAD

La matriz Jacobiana, ecuación (2), depende fuertemente de la posición del punto en la imagen, por lo que la suposición de que es constante solo será válida en el entorno del punto en la imagen. Por tal motivo una primera posibilidad es la de potenciar positivamente los movimientos ya efectuados con un pequeño recorrido en las características de la imagen. Sin embargo estos movimientos son excesivamente sensibles al ruido, por lo que será necesario buscar un compromiso entre ambos aspectos. Igualmente parecen más adecuados aquellos movimientos ya efectuados que se producen en el entorno articular del movimiento deseado. Los algoritmos propuestos catalogan los movimientos ya efectuados según una fiabilidad que depende de dos factores, que recogen estos conceptos:

$$\text{fiabilidad}_{ki} = \text{Factor}1_i / \text{Factor}2_{ki} \quad (12)$$

Donde el subíndice k representa el último movimiento efectuado y el subíndice i variará entre los movi-

mientos ya efectuados que hayan sido almacenados. Es necesario destacar que el $\text{Factor}1_i$ depende sólo de los movimientos ya efectuados, mientras que en el $\text{Factor}2_{ki}$ también interviene el último movimiento efectuado.

El $\text{Factor}1_i$ potencia movimientos en los que las características varían dentro de un rango: no son muy grandes para que el Jacobiano se pueda considerar constante ni son muy pequeños para que no sean muy sensibles al ruido. La fórmula empleada es:

$$\text{Factor}1_i = \frac{l}{\frac{\text{abs}(\|\Delta \mathbf{s}_i\| - F)}{F} + R} \quad (13)$$

Donde R representa la estimación del error en el cálculo de características (por ejemplo, para 5 puntos detectados con precisión subpixel un valor cercano a 1.0 píxeles), y F representa una estimación del máximo incremento fiable en las características para considerar que el Jacobiano no cambia (por ejemplo 20 píxeles). Este factor no depende del último movimiento realizado, por lo que puede ser calculado y almacenado junto con el correspondiente movimiento.

El $\text{Factor}2_{ki}$ potencia aquellos movimientos ya almacenados que se han producido en el entorno articular del movimiento deseado, comparándose a tal fin la posición articular actual con el punto medio de las posiciones articulares ya efectuadas. Su expresión es:

$$\text{Factor}2_{ki} = \left\| \mathbf{q}_k - \left(\frac{\mathbf{q}_i + \mathbf{q}_{i-1}}{2} \right) \right\| \quad (14)$$

3.2 INCORPORACIÓN DE LA RESTRICCIÓN EPIPOLAR

Un punto en el espacio $\tilde{\mathbf{S}} = [X \ Y \ Z \ 1]^T$, donde \sim representa coordenadas homogéneas, se proyecta en la primera imagen en el punto $\tilde{\mathbf{s}}' = [x' \ y' \ 1]^T = \mathbf{P}' \tilde{\mathbf{S}}$, donde \mathbf{P}' representa la matriz de proyección (3x4) de la primera cámara. Este punto se proyectará en la segunda cámara en el punto $\tilde{\mathbf{s}}'' = [x'' \ y'' \ 1]^T = \mathbf{P}'' \tilde{\mathbf{S}}$ donde \mathbf{P}'' representa la matriz de proyección (3x4) de la segunda cámara. Los puntos se expresan en coordenadas homogéneas y se asume un modelo sin distorsión (que es fácilmente corregible si se conoce). La calibración euclídea de ambas cámaras implica el conocimiento de las matrices \mathbf{P}' , \mathbf{P}'' , algo que no se desea realizar en el algoritmo propuesto.

La proyección de un mismo punto sobre las dos imágenes debe de cumplir la relación:

$$\tilde{s}^T F \tilde{s} = 0 \quad (15)$$

donde F es una matriz (3x3) que se denomina matriz fundamental o restricción epipolar. Su conocimiento se conoce como calibración débil o calibración proyectiva, y su detección es mucho más robusta que la calibración euclídea. Una descripción más detallada se puede consultar en [6] y [4].

El método propuesto considera la restricción epipolar, ecuación (15), en el cálculo del Jacobiano de la imagen, ecuación (1). Si las características visuales son centroides de puntos, y un punto en la primera cámara se expresa por $'$, y un punto en la segunda cámara se expresa por $''$, se tendrá el siguiente modelo:

$$s_k = \begin{bmatrix} s'_k \\ s''_k \end{bmatrix} ; \quad s_{k-1} = \begin{bmatrix} s'_{k-1} \\ s''_{k-1} \end{bmatrix} ; \quad J = \begin{bmatrix} J' \\ J'' \end{bmatrix} \quad (16)$$

$$s'_k = s'_{k-1} + J' \Delta q_k ; \quad s''_k = s''_{k-1} + J'' \Delta q_k \quad (17)$$

$$\begin{bmatrix} s_{k-1}^T & 1 \end{bmatrix} F \begin{bmatrix} s'_{k-1} \\ 1 \end{bmatrix} = 0 ; \quad \begin{bmatrix} s_k^T & 1 \end{bmatrix} F \begin{bmatrix} s'_k \\ 1 \end{bmatrix} = 0 \quad (18)$$

Sustituyendo en la ecuación (18) los valores obtenidos en las ecuaciones (17) se llega a la expresión no lineal para las incógnitas (J' , J''):

$$\Delta q_k^T J'^T F J' \Delta q_k + \Delta q_k^T J''^T F J'' \Delta q_k + \begin{bmatrix} s_{k-1}^T & 1 \end{bmatrix} F \begin{bmatrix} s'_{k-1} \\ 1 \end{bmatrix} + \begin{bmatrix} s_k^T & 1 \end{bmatrix} F \begin{bmatrix} s'_k \\ 1 \end{bmatrix} = 0 \quad (19)$$

Tanto las ecuaciones lineales (2) como las ecuaciones no lineales (19) se han resuelto conjuntamente aplicando el método de Levenberg-Marquardt.

3.3 GLOSARIO DE ALGORITMOS PROPUUESTOS

En la implementación descrita en el presente artículo se han utilizado los siguientes algoritmos para el cálculo del Jacobiano de la imagen:

- 3LAST: Últimos tres movimientos efectuados. Como el sistema articular empleado posee tres grados de libertad, la ecuación (2) puede ser re-

suelta utilizando los datos de solo tres movimientos.

- 3+REL: Tres movimientos más fiables entre los diez últimos efectuados.
- 10LASTW: Últimos diez movimientos efectuados, ponderados por su fiabilidad. La ponderación se introduce multiplicando cada fila de la ecuación (2) por su correspondiente fiabilidad.
- 10+REL: Diez movimientos más fiables entre los efectuados.
- FUNDMAT: Diez movimientos más fiables entre los efectuados, con la incorporación de la restricción epipolar.
- BROYDEN: Método de Broyden
- RLS: Estimación recursiva mediante mínimos cuadrados.
- KALMAN: Estimación recursiva mediante el filtro de Kalman.

Los algoritmos 3+REL, 10LASTW, 10+REL y FUNDMAT son originales.

4 PRUEBAS EFECTUADAS

En el presente apartado se describe primeramente el entorno de pruebas empleado y posteriormente los resultados obtenidos.

4.1 EQUIPO EXPERIMENTAL

El sistema utilizado en la realización de las pruebas consta de:

- Un sistema articular compuesto por un posicionador de alta precisión y su controlador, modelo Newport MM3000 (ver Figura 1). Posee tres grados de libertad que son dos ejes de giro y un eje de desplazamiento, y su precisión teórica es de la milésima de grado y la milésima de milímetro. Sobre el último eslabón del sistema articular y en un plano se han dispuesto el objeto visual a controlar, formado por cinco puntos negros sobre fondo blanco, y cuya proyección en la imagen serán las características empleadas en el control.
- Un sistema de adquisición y procesamiento de imágenes compuesto por dos cámaras analógicas CV-M50 y una tarjeta de adquisición de imágenes Meteor II-MC de Matrox, que permite la adquisición simultánea de las dos cámaras. Las cámaras, fijas en el espacio de trabajo, están separadas 642 milímetros, poseen los dos ejes convergentes hacia el sistema articular, y distan del mismo unos 900 milímetros. La detección de las características visuales se realiza con precisión subpíxel, y dada la sencillez de la imagen se estima que con un error menor de 0.2 píxeles. La comunicación con el controlador del sistema articular se realiza mediante un línea serie RS-232C.

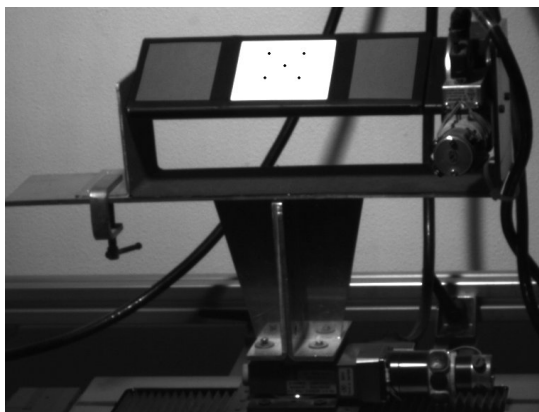


Figura 1: Entorno de pruebas

Es necesario destacar que el sistema articular es un sistema cerrado: una vez seleccionada una posición, no devuelve el control hasta que la alcanza. Esta forma de trabajo es inaceptable para un sistema de control visual que debe utilizar la información suministrada por las cámaras lo antes posible. Para simular un comportamiento más acorde, la opción elegida ha sido limitar el movimiento articular, en función del tiempo estimado que tardaría el sistema de visión en adquirir y procesar las imágenes. Por el contrario, la alta precisión del sistema permite mejorar la comparación de los métodos propuestos.

4.2 OBJETIVO DE CONTROL

La tarea encomendada al sistema es lograr el seguimiento de una trayectoria previamente fijada (ver figura 2), a partir de la información obtenida en la detección de un cierto número de características (centroides de la proyección de puntos). Se pretende contrastar las prestaciones de los métodos propuestos en la estimación del Jacobiano usando la ley de control del apartado 2.3.

Las características visuales deben de ser realizables y el objeto a controlar debe de ser visible desde ambos puntos de vista. Para asegurar esta coherencia, se ha optado por obtener previamente las características visuales, adquiriendo las imágenes en posiciones articulares de referencia, obtenidas mediante la generación de curvas alabeadas con parámetros aleatorios. Esta forma de generar las características se la conoce como “teach-by-showing”. Debido a la dependencia que presenta el Jacobiano con respecto a las características visuales y al punto en la espacio articular, se ha optado por encadenar un elevado número de trayectorias (en concreto 50), para obtener resultados más representativos en el estudio comparativo de los algoritmos propuestos.

4.3 ÍNDICES DE EVALUACIÓN

Para evaluar la efectividad de cada método, se consideran 3 índices, que se definen como:

- Índice 0: La suma de las distancias euclídeas entre las características visuales actuales y las deseadas. Está ponderado por el número de puntos, el número de cámaras y el número de trayectorias.
- Índice 1: La suma de las distancias euclídeas en el espacio articular para el conjunto de movimientos efectuados, dividido por mil. Está ponderado por el número de trayectorias.
- Índice 2: Suma de distancias euclídeas entre las posiciones articulares deseadas y las obtenidas. Está ponderado por el número de puntos.

4.4 RESULTADOS OBTENIDOS

Se ha realizado un estudio comparativo entre los algoritmos propuestos para la estimación del Jacobiano. La comparación abarca tanto a características visuales cuyo cálculo se estima con un error menor de 0.2 píxeles, por lo que se considerarán sin ruido, como características visuales a las que se les añade artificialmente un ruido gaussiano con una desviación típica de 0.5 píxeles. Igualmente se analiza la influencia de incrementar el número de puntos o características visuales de 2 a 5.

Así en la

Tabla 1 se recogen los resultados obtenidos para todos los algoritmos, sin o con ruido, en cada uno de los índices definidos, y variando el número de puntos empleados. Los métodos que obtienen mejores resultados son los de KALMAN, RLS, FUNDMAT y 10LASTW. Añadiendo ruido, el más robusto de todos es el FUNDMAT, especialmente en el índice 2, mientras que RLS y KALMAN también muestran un buen comportamiento. Incrementar el número de puntos tiene dos efectos opuestos: Se posee más información para controlar el sistema, pero también es más difícil cumplir plenamente el objetivo propuesto, por lo que causa un incremento en los índices.

La Figura 2 (FUNDMAT), la Figura 3 (RLS) y la Figura 4 (KALMAN) representan la evolución en el espacio articular para cada método empleado en la estimación del Jacobiano. Los puntos rojos representan los puntos de referencia en la trayectoria generada, y la línea azul muestra la evolución del sistema articular con la ley de control basada en el Jacobiano. Se aprecia que tanto con ruido como sin ruido, el mejor seguimiento se realiza con el algoritmo FUNDMAT.

Tabla 1: Valor de los tres índices, en función del número de puntos, para los algoritmos considerados.

	ALGO-RITMO	SIN RUIDO				CON RUIDO			
		2 PUNTOS	3 PUNTOS	4 PUNTOS	5 PUNTOS	2 PUNTOS	3 PUNTOS	4 PUNTOS	5 PUNTOS
INDICE 0	3LAST	16.6	17.9	18.6	18.5	20.4	21.8	21.5	22.1
	3+REL	16.8	18.2	18.0	17.8	17.2	21.0	19.6	22.0
	10LASTW	16.5	17.5	17.9	17.7	18.4	17.8	20.7	19.9
	10+REL	16.5	17.7	17.9	17.7	18.5	20.9	19.4	21.4
	FUNDMAT	16.5	17.5	17.7	17.6	16.6	17.7	17.9	17.7
	BROYDEN	16.4	18.2	18.7	17.8	16.9	19.6	22.8	19.0
	RLS	16.5	17.7	18.1	17.7	16.7	18.8	18.4	18.4
INDICE 1	KALMAN	16.0	17.3	17.7	17.6	17.4	18.0	18.7	18.5
	3LAST	3.43	3.48	3.68	3.63	2.22	2.55	2.29	2.39
	3+REL	3.68	3.47	3.44	3.42	3.60	2.45	3.66	2.40
	10LASTW	3.40	3.37	3.37	3.37	3.39	3.57	2.92	3.40
	10+REL	3.37	3.32	3.34	3.35	3.18	2.42	3.44	2.49
	FUNDMAT	3.35	3.33	3.34	3.35	3.36	3.34	3.31	3.34
	BROYDEN	3.37	3.43	3.66	3.38	3.34	2.86	3.84	3.32
INDICE 2	RLS	3.34	3.31	3.38	3.34	3.44	3.59	3.49	3.44
	KALMAN	3.37	3.41	3.48	3.42	3.65	3.53	3.39	3.46
	3LAST	0.77	0.98	1.84	0.99	7.08	5.70	5.34	5.55
	3+REL	0.94	0.67	0.58	0.41	1.88	5.66	1.97	5.36
	10LASTW	0.59	0.33	0.39	0.34	4.11	0.78	4.99	3.65
	10+REL	0.46	0.69	0.53	0.37	4.14	6.01	2.97	5.14
	FUNDMAT	0.42	0.50	0.38	0.29	0.45	0.60	0.45	0.29
	BROYDEN	0.38	0.80	1.60	0.45	1.42	3.64	6.19	2.38
	RLS	0.50	0.43	0.58	0.34	1.17	2.81	1.06	1.17
	KALMAN	0.21	0.42	0.62	0.56	3.45	1.63	1.73	1.58

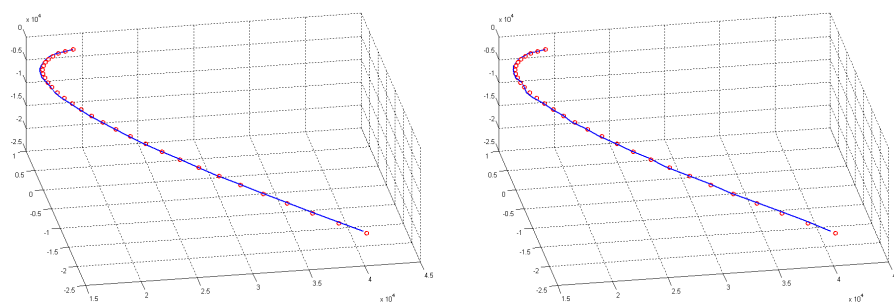


Figura 2: Evolución en el espacio articular para cinco puntos con el algoritmo FUNDMAT, sin y con ruido añadido

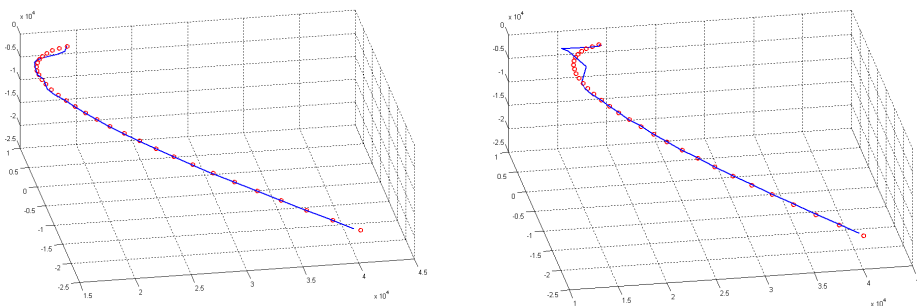


Figura 3: Evolución en el espacio articular para cinco puntos con el algoritmo RLS, sin y con ruido añadido

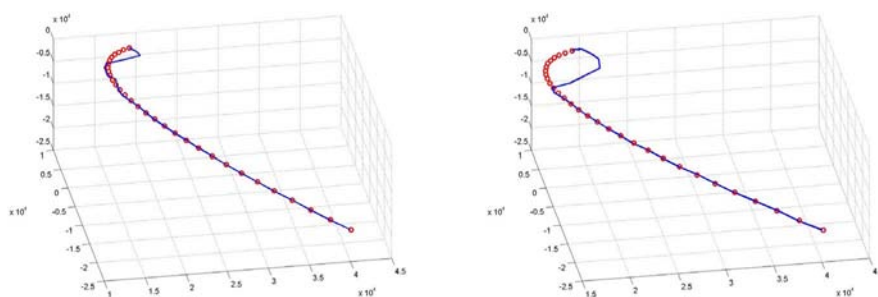


Figura 4: Evolución en el espacio articular para cinco puntos con el algoritmo KALMAN, sin y con ruido añadido

5 CONCLUSIONES

La estimación on-line del Jacobiano de la imagen es un método flexible y versátil para el control visual de una estructura articular pues aísla los resultados obtenidos de los errores producidos en la calibración de la cámara y en la calibración del sistema articular. Estos aspectos son especialmente críticos en muchas aplicaciones. Los algoritmos propuestos en el presente artículo introducen tanto el concepto de fiabilidad como la inclusión de la restricción epipolar en la estimación del Jacobiano. Este aspecto no ha sido tratado por otros autores, y aumenta la robustez de la estimación cuando las características se detectan con un moderado nivel de ruido. El conocimiento de la matriz fundamental no es una dificultad relevante, pues su cálculo es mucho más simple, robusto y fiable que la completa calibración de las cámaras y del sistema articular.

Algunos aspectos no tratados en el presente artículo y que están siendo estudiados en la actualidad es el análisis de la estabilidad del sistema con una ley de control generada a partir de la estimación del Jacobiano y el cálculo analítico del Jacobiano, para poder contrastar los resultados. Con respecto al primer aspecto es necesario destacar que en las numerosas pruebas efectuadas con ruido moderado, nunca el algoritmo FUNDMAT ha producido una inestabilidad en el sistema.

Este trabajo está subvencionado por la Comisión Interministerial de Ciencia y Tecnología, Proyecto DPI2004-07433-C02-02, y por la Agencia Española de Cooperación Internacional (AECI) dentro del Programa de Cooperación Interuniversitaria de 2005. También se enmarca en las actividades que el grupo desarrolla dentro del proyecto RoboCity2030 de la CAM, referencia S-0505/DPI/0176.

Referencias

- [1] Asada, M., Tanaka, T., Hosoda, K. (2000): Adaptive Binocular Visual Servoing for Independently Moving Target Tracking. ICRA 2076-2081.
- [2] Corke, P.I. (1996): Visual Control of Robots: High Performance Visual Servoing. Research Studies Press.
- [3] Deng, Z., Jägersand, M. (2004): Evaluation of Model Independent Image-Based Visual Servoing. 138-144.
- [4] Faugeras, O., Luong, Q.T. (2001): The Geometry of Multiple Images. MIT Press.
- [5] González, C. (2005). Estimación del Jacobiano de la Imagen para el Control Visual de un Sistema Articular no Calibrado. Proyecto Fin de Carrera de la E.T.S.I.I. de la Universidad Politécnica de Madrid
- [6] Hartley, R., Zisserman, A. (2000): Multiple View Geometry in Computer Vision. Cambridge.
- [7] Hutchinson, S.A., Hager, G.D., Corke, P.I. (1996): A tutorial on visual servo control. IEEE Trans. Robotics and Automation, 12-5 651-670.
- [8] Kragic, D., Christensen (2002): Survey on visual servoing for manipulation. Technical Report ISRN KTH/NA/P-02/01-Sen, CVAP259.
- [9] Kragic, D., Christensen, H.I. (2005). Advances in robot vision. Robotics and Autonomous Systems. 52 (1), 1-3
- [10] Lamiroy, B., Espiau, B., Andreff, N., and Horaud R. (2000). Controlling Robots With Two Cameras: How to Do it properly. Proceedings of the International Conference on Robotics and Automation, ICRA'00. 2100-2105.
- [11] Piepmeyer, J.A., McMurray, G.V., Lipkin, H. (2004): Uncalibrated Dynamic Visual Servoing. IEEE Transactions on Robotics and Automation, vol 20, nº1 143-147.
- [12] Quian, J., Su, J (2002). Online estimation of Image Jacobian Matrix by Kalman-Bucy filter for uncalibrated Stereo Vision Feedback. International Conference on Robotics & Automation (ICRA'02). 562-567
- [13] Sebastián, J.M, Pari, L., González, C., Ángel, L. (2005). A New Method for the Estimation of the Image Jacobian for the Control of an Uncalibrated Joint System. Lecture Notes in Computer Science, vol 3522, 631-638
- [14] Sutanto, H., Sharma, R., Varma, V.: The role of exploratory movement in visual servoing without calibration. Robotics and Autonomous Systems 23 (1998) 153-169.

Análisis de fuerzas en el mecanizado a alta velocidad

M. de J. Ramírez⁺, J. Pamies Texeira⁺⁺

M. Correa⁺⁺⁺, J.R. Alique⁺⁺⁺, Ciro A. Rodríguez G.⁺

⁺Instituto Tecnológico y de Estudios Superiores de Monterrey, Av. Eugenio Garza Sada N°2501 sur, Monterrey, Nuevo Leon, México C.P. 64849 {miguel.ramirez, ciro.rodriguez@itesm.mx}

⁺⁺Faculdade de Ciências e Tecnologia - UNL, Quinta da Torre, 2825 Monte Caparica, Portugal, {jorge.pamies@fct.unl.pt}

⁺⁺⁺Instituto de Automática Industrial, ctra. de Campo Real km. 0.200 La Poveda, Arganda del Rey, Madrid, España C.P. 28500 {macorrea, jralique@iai.csic.es}

Resumen

El conocimiento de la influencia de las fuerzas de corte es fundamental para el entendimiento a profundidad del comportamiento cinemático y dinámico de los procesos de mecanizado a alta velocidad. Los actuales sistemas de sensado de fuerzas durante el mecanizado, como las plataformas dinamométricas, solo son capaces de adquirir las fuerzas totales presentes lo que hace necesario un análisis para la separación de señales. En la literatura científica, es muy común analizar las fuerzas de corte cinemáticas como un promedio de los datos muestreados, sin hacer una separación real de las señales que son específicas de las fuerzas de corte. En el presente trabajo se muestran las herramientas necesarias y una metodología para realizar un análisis de señal que tome en cuenta la física del proceso más allá de un simple promedio de datos.

Palabras Clave: Fuerzas de corte, Mecanizado a alta velocidad, análisis de señal, FFT, modelación.

1 INTRODUCCIÓN

Debido a su propia naturaleza, en el mecanizado a alta velocidad predominan las fuerzas dinámicas sobre las fuerzas de corte, a diferencia del mecanizado convencional donde las fuerzas dinámicas llegan a ser despreciables (ver figura 1). Las fuerzas de corte son aquellas fuerzas propias del proceso de mecanizado que influyen en fenómenos del corte asociados a la deformación del material. Las fuerzas dinámicas están asociadas con la oscilación alrededor de una posición de equilibrio del sistema con un valor medio típico alrededor de cero, pero que adicionadas a las anteriores influyen de forma

determinante en fenómenos importantes del mecanizado a alta velocidad como el acabado superficial, desgaste de herramientas y en el retemblado.

Los estudios profundos y confiables del proceso de alta velocidad requieren un análisis y separación de señales correspondientes a las fuerzas que son específicas del corte. Las plataformas dinamométricas, que son de los sensores más confiables para medir las fuerzas durante un mecanizado, arrojan señales correspondientes a las fuerzas totales presentes, por lo que es poco conveniente utilizar las señales originalmente adquiridas para el análisis profundo del proceso de corte.

Lo expresado anteriormente, permite concluir que para estudiar y caracterizar, con una buena aproximación, la física del proceso es necesario el análisis y separación de las señales de fuerza colectadas por los sensores. Si se quieren desarrollar modelos se deben de tener en cuenta estas dos componentes y analizar físicamente cuales son los fenómenos que dependen de la fuerza de corte (baja frecuencia) y los fenómenos que dependen de las fuerzas dinámicas (alta frecuencia).

Generalmente, en la literatura científica son utilizadas las fuerzas de corte directamente como son adquiridas del sensor, incluyendo en ella efectos dinámicos que no son propios del proceso de corte. Esto sucede en muchas de las aplicaciones del mecanizado, analizando solo algunos ejemplos tenemos que: En [9] se desarrolla una modelación para microfresado basado en fuerzas adquiridas directamente del sensor; de la misma manera, en [2] se determinan los coeficientes de corte para taladrado utilizando la fuerza total obtenida del sensor; igualmente, en [7] se desarrolló un controlador

borroso para torneado basado en la medición de las fuerzas de corte bajo las mismas condiciones.

También existe la tendencia a desarrollar análisis utilizando el promedio de las fuerzas muestreadas como la componente de corte o estática de las fuerzas sensadas [8]. En [4], se utilizó la fuerza promedio durante el corte como un parámetro característico para monitorear el estado de la herramienta.

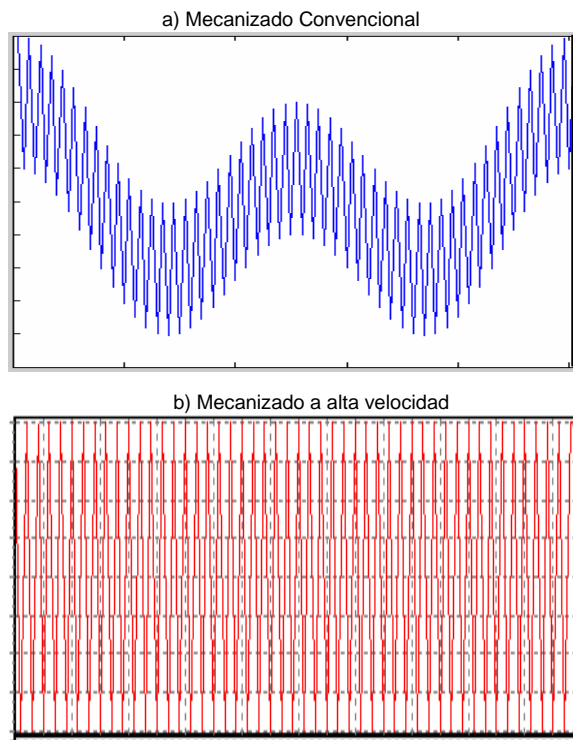


Figura 1: Mecanizado convencional vs Mecanizado a alta velocidad.

2 METODOLOGÍA

La metodología propuesta en el presente trabajo es mostrada en la Figura 2. Como ya se ha mencionado en el mecanizado a alta velocidad la señal de fuerza tiene una componente dinámica de magnitud mayor a la componente que representa la fuerza de corte. Para comprobar este hecho se realiza un primer análisis a la señal adquirida a través de la obtención de su espectro de Hilbert, donde se debe visualizar las frecuencias predominantes de la señal y su contenido energético. Típicamente, éstas deben ser la frecuencia de revolución, la frecuencia entre dientes y armónicas de éstas. Las ventajas que proporciona el espectro de Hilbert radican en el hecho de que permite análisis de señales no estacionarias y no lineales.

También se realiza un análisis frecuencial más común basado en la generación de un espectrograma para verificar las características fundamentales de la señal y así confirmar la información obtenida por el espectro de Hilbert.

Hasta esta parte la metodología permite de una forma concluyente verificar que las frecuencias dominantes son la frecuencia de revolución y la frecuencia entre dientes.

En una segunda etapa se debe de filtrar la señal original para separar las señales debajo de las frecuencias obtenidas en la etapa anterior que contienen las frecuencias del proceso de corte y las señales de alta frecuencia que contienen las frecuencias de los fenómenos dinámicos del proceso.

Una vez separadas las señales, en una tercera etapa se procederá a realizar su análisis FFT más detallado de las señales de corte y señales dinámicas, de esta manera se puede correlacionar los efectos de las variables del proceso.

2.1 ESPECTRO DE HILBERT y ESPECTROGRAMA

La mayoría de las técnicas utilizadas para el análisis espectral comúnmente aceptado están fundamentadas en la transformada rápida de Fourier [1][3]. Sin embargo, el análisis de series temporales presenta una serie de limitaciones debido a que el análisis de Fourier la señal es linealmente descompuesta en una combinación de senos y cosenos, y a que modela la señal de entrada como una señal estacionaria.

Por lo anteriormente expresado, los métodos que se fundamentan en la transformada de Fourier realizan un análisis de tiempo y frecuencia que pueden no resaltar la presencia de oscilaciones locales de la señal que pueden ser importantes [6].

Muchos fenómenos naturales pueden ser aproximados a ser representados por sistemas lineales y ser estacionarios. Para verificar si el fenómeno de corte es estacionario o no, se propone realizar un espectrograma el cual es una representación visual de las variaciones de la frecuencia en el eje vertical, y de la intensidad mediante niveles de colores que se está representando a lo largo del tiempo en el eje horizontal.

El espectro de Hilbert es una técnica que nos permite tratar aquellas señales que son no-lineales y no estacionarios. El espectro de Hilbert provee un contexto con los cuales se formalizan y generalizan los conceptos de las series de Fourier en términos de polinomios ortogonales arbitrarios y la propia transformada de Fourier [6].

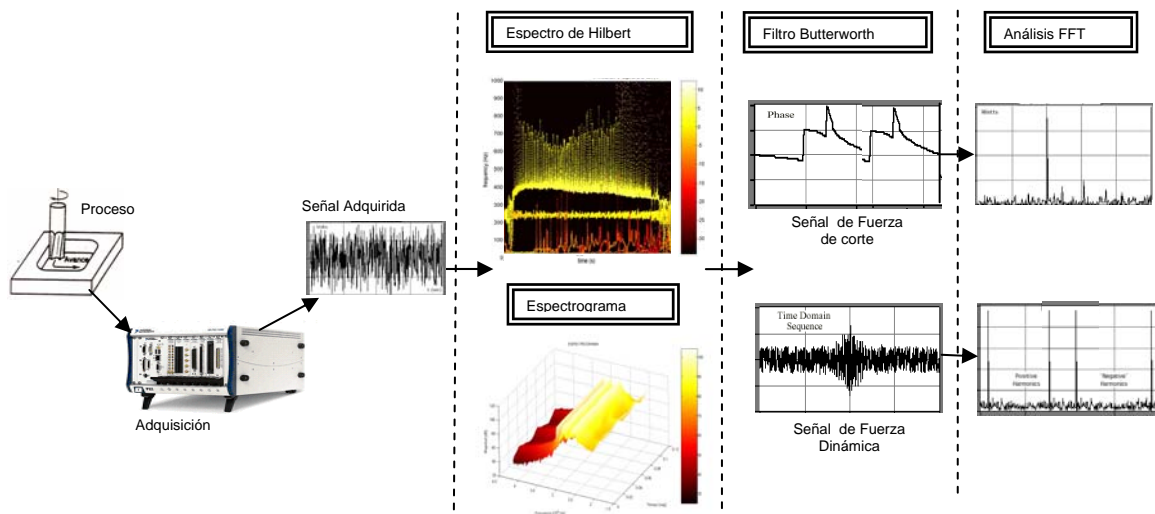


Figura 2: Esquema de la metodología

La generación del espectro de Hilbert (HS) se realiza en dos fases. En primer lugar, en el proceso llamado descomposición empírica de modo (Empirical Mode Decomposition, EMD) en que se descompone la serie temporal de entrada en un conjunto de funciones llamadas funciones modales intrínsecas (Intrinsic Mode Function, IMF) que representan modelos de fenómenos oscilatorios correspondientes a las frecuencias dominantes. Este proceso identifica automáticamente la escala de tiempo apropiada para el estudio de las características físicas de la señal estudiada, [5]. A continuación, se aplica la llamada transformada de Hilbert para la descomposición de las funciones modales intrínsecas (IMF) y la construcción de la distribución de la energía en el dominio del tiempo-frecuencia ó denominado espectro de Hilbert.

En el presente documento se propone utilizar el espectro de Hilbert como una herramienta exploratoria para la identificación de las características fundamentales de la señal generada en el proceso de mecanizado a alta velocidad.

2.2 TÉCNICAS DE SEPARACIÓN DE SEÑALES

Una vez que se tiene determinado las características principales de nuestra señal, se procede a realizar la separación de señales propias de corte como de las señales propias de la dinámica del proceso.

Para el filtraje de la señal se utiliza el filtro de Butterworth, que es uno de los más utilizados, debido a que produce menos alteración de la señal en la

frecuencia de corte del filtro. En el caso de las señales del mecanizado a alta velocidad, la frecuencia de corte del filtro será aquella que se obtenga del análisis realizado en la etapa anterior de la metodología. Esta frecuencia de filtraje deberá de ser el límite que separa las frecuencias de corte de las frecuencias dinámicas, o sea, la frecuencia entre dientes para el caso del proceso fresado. Al final de esta etapa de la metodología tendremos dos señales separadas: una que representa el proceso de corte puro y otra que representa las características dinámicas del mismo.

2.3 ANÁLISIS EN FRECUENCIA

La tercera etapa de la metodología es el análisis en frecuencia de cada uno de los tipos de señales obtenidos hasta el momento. Para ello se propone utilizar el análisis de Fourier (FFT). Las series de Fourier describen señales periódicas como una combinación de señales armónicas (sinusoides) y permite analizar una señal oscilatoria estacionaria en términos de su contenido frecuencial de tal forma que operaciones realizadas en el dominio temporal tienen su dual en el dominio frecuencial.

3 EXPERIMENTACIÓN

Para demostrar la metodología propuesta, se realizó una etapa experimental basada en un diseño de experimentos factorial fraccionado utilizando 8 factores a dos niveles ($2^{(8-3)}$) como variables independientes y la rugosidad superficial como variable de respuesta.

Se realizaron un total de 32 experimentos con aluminio de diferentes durezas, en la tabla 1 se muestran las variables y el valor de los respectivos niveles utilizados.

Nivel	Código	fz (mm/diente)	Vc (m/min)	ap (mm)	Dtool (mm)	ae (mm)	Dureza (HB)	Radio (mm)	I/C
Alto	1	0	850	10	16	5	145	40	exterior
Bajo	-1	0	500	5	12	1	65	20	interior

Donde :

fz = Avance por diente.

Vc = Velocidad de corte.

ap = Profundidad axial.

Dtool = Diámetro de la herramienta.

ae = Profundidad radial.

Dureza = Dureza Hard Brinel

I/C = Geometría de pieza.

Tabla 1: Valores de los niveles de las variables utilizadas en el diseño de experimentos.

Los experimentos fueron realizados en un centro de mecanizado Kondia HS1000 equipado con un CNC abierto marca Siemens modelo Sinumerik 840D. Para la medición de las fuerzas de corte se montó en la mesa de la máquina una plataforma dinamométrica marca Kistler modelo 9257B.

4 RESULTADOS

Para propósitos del presente trabajo, se mostrará la aplicación de la metodología propuesta a la componente X de la fuerza medida por la plataforma dinamométrica para ejemplificar su uso. El experimento para generar esta fuerza fue realizado en una probeta con una geometría tipo caja circular con una avance por diente de 0.13 mm/diente, una velocidad de corte de 500 m/min, una herramienta para fresar de 12 mm de diámetro, la pasada radial fue de 1 mm y la pasada axial de 10 mm. El material utilizado fue un aluminio 5083.

Del análisis de Hilbert se obtuvo la gráfica para la señal (figura 3), de donde se puede concluir que existen dos frecuencias dominantes, una que representa la frecuencia de rotación de la herramienta (221 Hz) y otra que representa la frecuencia entre dientes (442 Hz). Se verifica también que estas frecuencias son estacionarias a lo largo del tiempo. Además, se visualiza que existen otras frecuencias de valor mayor y de contenido energético relevante que representan los fenómenos dinámicos antes referidos.

Como el método de Hilbert no está todavía validado para las señales de un proceso de corte a alta velocidad se realizó el espectrograma de la figura 4 que confirma exactamente los resultados

mencionados en el párrafo anterior. En el se visualiza que las frecuencias dominantes son estacionaria a lo largo del tiempo y también muestra los niveles de magnitud de cada frecuencia.

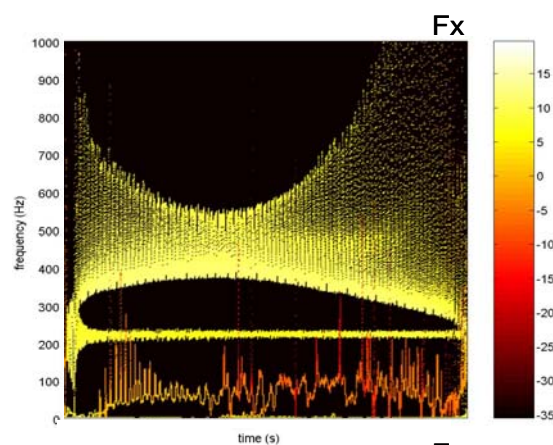


Figura 3: Espectro de Hilbert de la componente Fx.

En la figura 5 se muestra el resultado de la separación de la señal original en la componente de corte y en la componente dinámica. Se verifica en la figura que el promedio de (a) es -62 N que corresponde a la fuerza de corte convencional y en (b) el promedio es cero que significa que es la oscilación del sistema alrededor de su posición de equilibrio. Esto confirma que el valor de las fuerzas dinámicas se superpone a las fuerzas de corte en el mecanizado a alta velocidad.

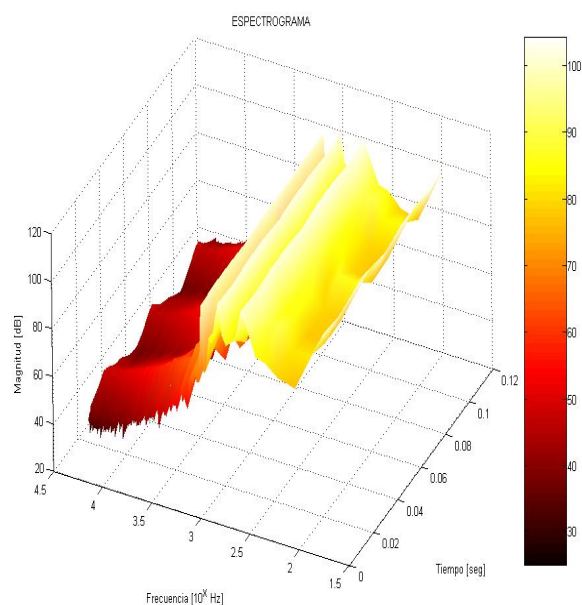
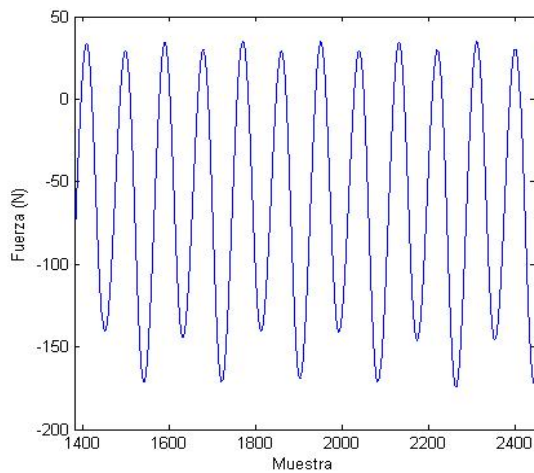
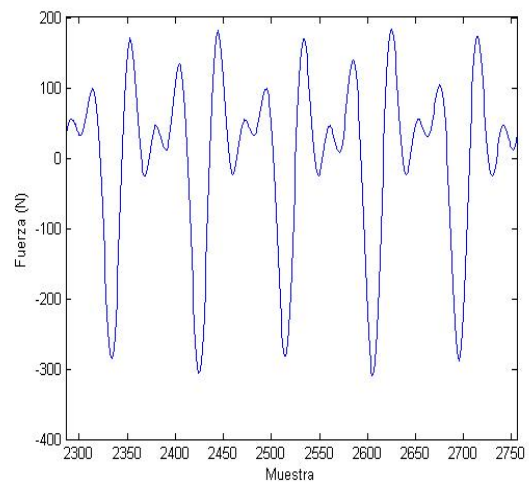


Figura 4: Espectrograma.



(a) Componente de la fuerza de corte

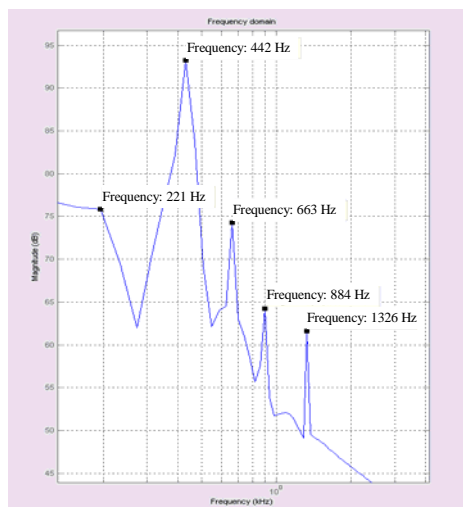


(b) Componente dinámica de la fuerza

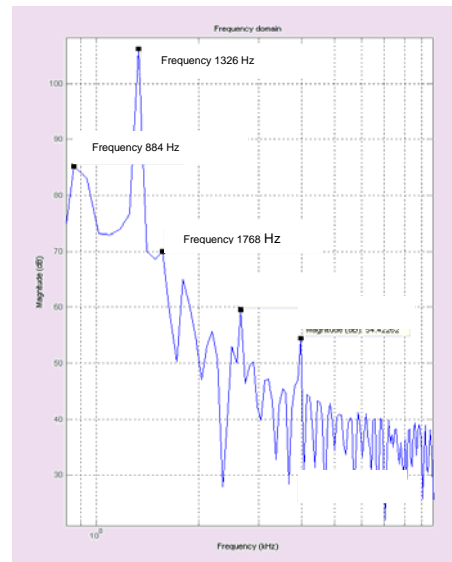
Figura 5 : separación de la componente X de la fuerza sensada.

El resultado del análisis frecuencial de las señales separadas que se muestra en la figura 6 nos indica que:

- En (a), verificamos que existen 3 frecuencias dominantes correspondientes a la frecuencia de revolución (221 Hz), frecuencia entre dientes (442 Hz) y una tercera armónica de la primera.
- Se denota también en la gráfica que la 4ª y 6ª armónica están presentes, pero que con respecto a las anteriores son de menor relevancia, sin embargo, éstas son frecuencia dominantes en la señal de más alta frecuencia mostrada en (b). Esto se debe a una cierta incertidumbre por el la frecuencia de corte del filtro Butterworth utilizado.
- En (b) se verifica también la aparición de una octava armónica con un nivel de energía también relevante, y el resto de las armónicas ya no tienen un gran significado físico.
-



(a)



(b)

Figura 6: Análisis en frecuencia (FFT)

5 CONCLUSIONES

Con este trabajo se puede concluir de una forma clara que:

- La señal original obtenida directamente del sensor, no tiene ningún significado físico si es considerada como fuerza de corte.
- Las fuerzas dinámicas en mecanizado a alta velocidad son de mayor intensidad (en un factor de 4 a 1) con respecto a la fuerza de corte efectiva.

Por otra parte, la metodología que se propone se ha mostrado capaz de proporcionar un medio para efectuar la separación de las señales consistentes con el fenómeno físico del corte.

Se necesita un trabajo adicional para una validación más profunda, comparando las fuerzas de corte obtenida por las señales adquiridas por los sensores con las fuerzas de corte calculadas a través de modelos mecánicos ya comprobados.

El presente trabajo propone que si no se realiza un análisis previo de la fuerza puede llevar a errores de interpretación debido a la falta de una correspondencia del análisis con los fenómenos físicos del proceso a bajas y altas frecuencias.

Agradecimientos

Los autores agradecen a la cátedra en investigación en Mecatrónica del Instituto tecnológico y de Estudios Superiores de Monterrey, México, por su apoyo en la realización de este trabajo.

De la misma manera, los autores reconocen el apoyo brindado por el programa de apoyo a la investigación denominado I3P perteneciente al Centro Superior de Investigaciones Científicas del gobierno Español.

Referencias

- [1] Gresty, M., Buckwell, L., (2000), "Spectral analysis of tremor: understanding the results.", *J Neurol Neurosurg Psychiatry*, 53, p.p. 976–981.
- [2] Hamadea, R., Seifa, C., Ismail, F., (2006), "Extracting cutting force coefficients from drilling experiments", *IJ of Machine Tools & Manufacture*, 46, p.p. 387-396.
- [3] Kay, S., Marple, S., (1981), "Spectrum analysis - a modern perspective", *Proceedings of the IEEE*, 69(11), p.p. 1380–1419.
- [4] Kuljanic, E., Sortino, M., (2005), "TWEM, a method based on cutting forces—monitoring tool wear" in face milling, *IJ of Machine Tools & Manufacture*, 45, p.p. 29-34.
- [5] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, y Henry H. Liu., (1998) "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, *Procedures of Royal Society of London*, 454, p.p. 903–995.
- [6] Rocon, E., (2006) "Reducción activa de temblor patológico de miembro superior mediante exoesqueletos robóticos", Tesis Docotral, Universidad Politécnica de Madrid. España.
- [7] Ruey-Jing L.,*, Bai-Fu L., Jyun-Han H., (2005), "A grey prediction fuzzy controller for constant cutting force in turning", *IJ of Machine Tools & Manufacture*, 45, p.p. 1047-1056.
- [8] Toh, C.K., (2004), "Static and dynamic cutting force analysis when high speed rough milling hardened steel", *Material & Design*, U.K. p.p. 41-50.
- [9] Zaman, M., Senthil, A., Rahman, M., Sreeram, S., (2006), "A three-dimensional analytical cutting force model for micro end milling operation". *IJ of Machine Tools & Manufacture*, 46, p.p. 353-366.

MODELO CONCEPTUAL DE LAS TAREAS DE ANÁLISIS Y DISEÑO DE COMPENSADORES EN EL LUGAR DE LAS RAÍCES

Isaías García

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, isaias.garcia@unileon.es

Carmen Benavides

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, carmen.benavides@unileon.es

Héctor Aláiz

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, hmoreton@unileon.es

Javier Alfonso

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, javi@unileon.es

Ángel Alonso

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, hmoreton@unileon.es

Francisco Rodríguez

Departamento de Ingeniería Eléctrica y Electrónica, Universidad de León, diefrs@unileon.es

Resumen

El presente artículo describe la creación de un modelo conceptual que refleja el conocimiento existente en el dominio de las técnicas de análisis y diseño de sistemas en el lugar de las raíces. Para la creación de esta estructura se utilizan las técnicas de modelado del conocimiento provenientes del campo de la Ingeniería del Conocimiento. También se describe una aplicación que hace las veces de tutor inteligente para este dominio utilizando esa representación del conocimiento.

Palabras Clave: Educación en control, Ontologías, Sistemas Basados en el Conocimiento. Software para control

1 INTRODUCCIÓN

Las aplicaciones informáticas actuales necesitan estructuras que reflejen de forma más adecuada el conocimiento de un dominio. Este hecho está provocado por la necesidad de aumentar el nivel del procesamiento de la información en el ordenador ante la creciente informatización e interconexión de los sistemas informáticos. La orientación a objetos y sus herramientas relacionadas no son capaces de expresar toda la riqueza conceptual que se exige al software bajo estas premisas. En la actualidad están apareciendo aproximaciones basadas en la creación de modelos del conocimiento como base para obtener la solución a estas necesidades. El mundo de la empresa, con la denominada gestión del

conocimiento, junto con la nueva idea del servicio web, la Web Semántica, son los campos donde con más fuerza se está sintiendo esta tendencia.

Las ideas que subyacen a esta nueva aproximación provienen del campo de la Ingeniería del Conocimiento [5] y más concretamente de la representación del conocimiento. La aproximación de esta disciplina consiste en la creación de estructuras formales de representación (o lo que es lo mismo, modelos) del conocimiento que conforman la base de las aplicaciones informáticas. Esta aproximación es radicalmente opuesta a la utilizada hasta el momento en el campo de la orientación a objetos, donde los modelos que se construyen son utilizados como herramientas de ayuda en la fase de diseño, desapareciendo una vez que se ha generado el código.

En el campo del Control Automático durante los últimos años han cobrado especial relevancia la necesidad de incrementar la interactividad entre el usuario y las aplicaciones informáticas en este dominio. Este hecho se manifiesta sobre todo en el campo del diseño asistido por ordenador y en el de la educación en control. A partir de finales del siglo XX han ido apareciendo diversas aplicaciones (ver por ejemplo [3] ó [2]) que hacen uso de las técnicas de orientación a objetos, capacidades multimedia y las comunicaciones a través de Internet para crear aplicaciones con gran interactividad y uso extensivo de representaciones gráficas así como para construir laboratorios virtuales y/o remotos. Estas aplicaciones han supuesto un gran avance comparadas con las existentes hasta el momento, de las que usualmente

se obtenían resultados numéricos sin ningún tipo de interactividad.

Estas nuevas aplicaciones interactivas emergentes pueden ver aumentada su utilidad y usabilidad utilizando las técnicas de las que se ha hablado en párrafos anteriores. Esto permitirá obtener una serie de beneficios en la interacción hombre-máquina, entre los que se pueden citar:

- El usuario pasará de interactuar con los datos y las graficas a interactuar con los conceptos asociados a esos datos o representaciones gráficas.
- Se podrá mantener el enlace conceptual entre las representaciones de los datos y los conceptos matemáticos y teóricos que subyacen a los mismos, hecho este que es un problema en una herramienta altamente interactiva con un nivel de representación del conocimiento de poca abstracción.

El presente artículo describe la creación de una estructura formal de representación del conocimiento (también llamado modelo conceptual) que refleja el conocimiento existente en un dominio del Control Automático, más en concreto en el dominio de las técnicas de análisis y diseño de sistemas en el lugar de las raíces. Esta estructura puede servir para construir aplicaciones de tipo CACE (Computer-Aided Control Engineering), tutores inteligentes o para crear aplicaciones web semánticas para el dominio del Control, con las ventajas citadas en el párrafo anterior.

El artículo está organizado del siguiente modo: En la sección 2 se presentan las ideas de la Ingeniería del Conocimiento así como los elementos constituyentes y la forma de crear y usar las estructuras o modelos conceptuales. En la sección 3 se presenta el dominio de estudio dentro de la Ingeniería de Control que se ha escogido y el modelo conceptual que se ha construido de ese dominio. Finalmente, en la sección 4 se presenta una discusión de los resultados obtenidos, describiendo brevemente una aplicación realizada para comprobar el uso de la estructura conceptual. La sección 5 está dedicada a posibles aproximaciones y trabajos futuros.

2 EL MODELADO DEL CONOCIMIENTO

La Ingeniería del Conocimiento es una rama de investigación enmarcada dentro de la Inteligencia Artificial que se encargó de recoger el fracaso de los Sistemas Expertos de los años 80 y estudiar desde un punto de vista más teórico las posibles soluciones a los problemas que provocaron ese fracaso.

Como consecuencia de los estudios llevados a cabo en esta disciplina a lo largo de los años se ha llegado a ciertos puntos de acuerdo sobre la forma en la que abordar la representación del conocimiento [4]:

- Necesidad de estudiar la representación del conocimiento desde un punto de vista formal.
- Separar la descripción del conocimiento de su implementación. Una estructura conceptual solamente describe el conocimiento de un dominio. Idealmente una representación del conocimiento no debería presuponer siquiera qué tipo de agente usará esa descripción.
- Considerar uno de los objetivos más importantes de la creación de estructuras conceptuales la reutilización de las mismas en diferentes aplicaciones. Las estructuras conceptuales deben tener mecanismos para ser evaluadas, extendidas o fusionadas con otras.

En cuanto a la estructura conceptual, un modelo del conocimiento puede dividirse en dos partes:

- La estructura estática. Recoge los conceptos y relaciones existentes en ese dominio. Los conceptos pueden considerarse como los nombres y adjetivos existentes en la descripción del dominio. En cuanto a las relaciones, éstas pueden ser de muy diversa índole: estructurales, topológicas, mereológicas, teleológicas, etc. Las propiedades se modelan como conceptos especiales y tendrán también características asociadas (transitividad, reflexividad, etc).
- Estructura dinámica. Describe la estructura de tareas y subtareas en las que se puede dividir la descripción de las estrategias de resolución de problemas que se dan en el dominio de estudio.

La estructura conceptual que recoge estos tres componentes básicos (conceptos, relaciones y tareas) se denomina ontología [1](Gruber, 1995) y constituye hoy en día uno de los campos de investigación más activos dentro de la Ingeniería del Conocimiento e incluso dentro de la Inteligencia Artificial en su conjunto.

La creación de un modelo conceptual tiene sentido cuando el dominio en base al cual se crea ese modelo presente gran contenido semántico, es decir, gran cantidad de conceptos y relaciones que conformen una compleja red. En estos casos el uso de técnicas tradicionales de orientación a objetos haría que esa estructura de conocimiento se perdiese y no pudiese ser utilizada durante la ejecución de la aplicación. Otra de las condiciones que hace que sea beneficioso usar esta aproximación es que se quiera que el

programa interactúe con el usuario de forma muy flexible y a alto nivel y sea capaz de ofrecer información sobre la estructura de su conocimiento interno así como los procesos que han llevado a una determinada solución.

3 UN MODELO CONCEPTUAL PARA EL LUGAR DE LAS RAÍCES

A la hora de crear un modelo conceptual es necesario encontrar un dominio acotado de conocimiento al que aplicar estas técnicas. En el caso del presente artículo se ha elegido como dominio las técnicas de análisis y diseño de controladores (de adelanto-retraso de fase) en el lugar de las raíces.

3.1 EL MÉTODO DEL LUGAR DE LAS RAÍCES

En control automático tienen gran importancia los métodos gráficos para la representación de los problemas de análisis y diseño. Las dos principales formas de estudiar el comportamiento de un sistema y abordar el diseño del mismo dentro de la teoría clásica del control son la respuesta temporal y la respuesta en frecuencia. En cuanto a herramientas gráficas, la más relacionada con la respuesta temporal es el diagrama del lugar de las raíces, mientras que en el caso de la respuesta en la frecuencia se cuenta con diversos diagramas como el de Bode, el de Nyquist o el de Nichols. Los métodos de la respuesta en frecuencia han ido ganando terreno a la hora de realizar el diseño de un sistema de control debido a la mayor relevancia que han ido cobrando temas como el diseño para el rechazo de perturbaciones y ruidos introducidos en el sistema o técnicas relativamente nuevas como el control robusto.

Sin embargo, la respuesta temporal sigue siendo la más intuitiva para el ser humano y la que más información nos da sobre si un determinado diseño tiene un comportamiento acorde con las especificaciones impuestas de partida al reflejar de forma directa el comportamiento del sistema durante el régimen transitorio cuando se produce un cambio en la señal de referencia o una variación en la carga del proceso. La relación entre la respuesta temporal y las herramientas gráficas citadas anteriormente puede establecerse en todos los casos aunque en el caso de la representación de las raíces en el plano complejo y el lugar de las raíces esta relación es más directa e intuitiva.

Observando la posición de los polos y ceros del modelo en función de transferencia de un sistema en el plano complejo se puede obtener una idea bastante aproximada de cómo es la forma de la respuesta

temporal del sistema ante una variación de tipo escalón en la señal. El tipo de raíz (real o compleja) de los polos del sistema nos hablan de la forma de la onda resultante, mientras que el valor numérico y la posición en el plano de las partes real y compleja de esas raíces están relacionadas directamente con la rapidez de la respuesta y la frecuencia de oscilación de la onda de salida respectivamente, por poner un ejemplo. En cuanto al diseño, la representación del lugar de las raíces es una herramienta que pone rápidamente de manifiesto el tipo de control a implementar para conseguir unas especificaciones dadas en la respuesta a un escalón así como la influencia que la variación en los parámetros del controlador que se diseña tendrá en la posición de polos y ceros del sistema en cadena cerrada y por tanto en esa respuesta temporal del sistema total.

Por estas razones, a pesar de que en la práctica se suelen emplear las técnicas de la respuesta en frecuencia para hacer el diseño de controladores, se ha preferido a la hora de tener una experiencia en la creación de un modelo conceptual utilizar los métodos del lugar de las raíces, que se encuentran semántica o conceptualmente más cercanos a los conceptos de respuesta en el tiempo.

3.2 EL MODELO CONCEPTUAL

Seguidamente se presentan una serie de ejemplos de cómo se ha construido el modelo conceptual para el dominio elegido. Como se vio en la sección 2, la creación de un modelo conceptual consiste en la búsqueda de los conceptos, relaciones y tareas presentes en el dominio.

3.2.1 Modelado de conceptos y relaciones

Los métodos relacionados con el lugar de las raíces están incluidos dentro de la teoría clásica de control y por lo tanto utilizan modelos de representación en el dominio de la frecuencia compleja, en el “dominio s”. Estos modelos están caracterizados por la denominada función de transferencia, una estructura matemática algebraica consistente en un cociente de polinomios que representan la relación entre las transformadas de Laplace de la señal de salida y la correspondiente a la de la señal de entrada. Usando esa representación y las características asociadas a la misma se efectúa todo el razonamiento en este campo. Por lo tanto el concepto de función de transferencia y todos los conceptos asociados al mismo son la base del modelo conceptual.

Estructuralmente una función de transferencia está compuesta por un cociente de polinomios: uno de ellos en el papel de numerador y el otro en el de denominador. Cada uno de estos polinomios serán conceptos a su vez dentro del modelo conceptual y

estructuralmente estarán formados por sus descripciones que serán bien en forma de coeficientes de potencias decrecientes o bien una serie de raíces más una ganancia. En el caso de la descripción en potencias decrecientes, los términos serán números reales (condición que se cumple al tratarse de sistemas físicos). En el caso de la descripción en forma de lista de raíces y ganancia tendremos por un lado una serie de números complejos (representando a las raíces) y un número real para representar a la ganancia.

Vemos que la definición de conceptos en cuanto a su estructura exige la representación de diferentes conceptos (función de transferencia, cociente, polinomios, número real, número complejo, etc). Además de estos conceptos, se deberá especificar cómo están relacionados unos con otros a la hora de construir los de más alto nivel a partir de los de más bajo nivel. De esta forma habrá que representar la relación de tipo estructural que define de qué están compuestos los conceptos. Habitualmente este tipo de relaciones estructurales se representan como propiedades asociadas directamente a los conceptos que los definen. Este tipo de relación estructural es tan importante que muchas herramientas de creación de ontologías proporcionan un mecanismo de representación para las mismas. Este mecanismo es el denominado slot, un término que proviene de la definición de marco como estructura de representación de conocimiento (Newell 84). Un slot es por tanto una relación estructural entre un concepto y otros conceptos que definen su estructura. Cabe hacer notar que en orientación a objetos se tomó esta idea como base para crear las denominadas “propiedades” de un objeto, con la salvedad de que allí estas propiedades no tienen la restricción de expresar una relación estructural, de composición, sino que pueden utilizarse para decir cualquier cosa del objeto.

En el caso que nos ocupa los slot tienen este valor estructural, es decir, definen características estructurales para el concepto que los contiene. A modo de ejemplo, para el concepto función de transferencia un posible slot será el que relaciona este concepto con un cociente de polinomios. En el ejemplo que nos ocupa esto se ha modelado introduciendo dos conceptos: `TransferFunctionSystemModel` y `PolynomialQuotient` y creando un slot asociado al concepto `TransferFunctionSystemModel` denominado “`hasPolynomialQuotient`” que se define como una instancia concreta del concepto `PolynomialQuotient`.

Los slot por tanto especifican relaciones estructurales de composición. El resto de características que no son de tipo estructural se modelan aparte,

separándolas del concepto principal. A modo de ejemplo, para un concepto `TransferFunctionSystemModel` podríamos tener una posible característica asociada al mismo que fuese “orden”, ya que es habitual hablar del orden del modelo del sistema. Sin embargo esta característica no es estructural, es decir, no define una relación de composición respecto al concepto al que se asocia, se puede prescindir de ella y el concepto sigue teniendo validez y estando definido. El concepto “orden” es simplemente una característica o algo que se puede decir de un concepto `TransferFunctionSystemModel` pero no entra dentro de la definición del concepto y por tanto no se debe modelar como un slot. En el caso de la orientación a objetos, todo lo que se puede decir de un objeto en una aplicación se modela como una propiedad del mismo, sin tener en cuenta esta distinción entre característica estructural o no. Esta es una de las diferencias entre un modelo conceptual del conocimiento y un modelo en orientación a objetos.

Por seguir con el ejemplo del orden de un sistema, este tipo de características no estructurales se modelan aparte tal como se ha mencionado, en el caso de este ejemplo se han modelado como características cuantitativas, más en concreto como características cuantitativas y relacionadas con el estudio de sistemas en el dominio de la frecuencia compleja. Esto sugiere que las cosas que se pueden decir de un sistema (características) pueden organizarse conceptualmente en una jerarquía, lo cual sirve para introducir otro tipo de relación, la relación “subtipo” ó relación “is-a”. De este modo, las características pueden dividirse en cualitativas y cuantitativas, o bien en las relacionadas con el dominio de la frecuencia (teoría clásica de control) o del dominio del tiempo (teoría moderna de control). Incluso las relacionadas con el dominio de la frecuencia compleja pueden dividirse entre las relacionadas con la respuesta en el tiempo y las relacionadas con la respuesta en frecuencia. A su vez éstas pueden dividirse aún más utilizando esta relación. Por ejemplo, las características relacionadas con la respuesta en el tiempo pueden dividirse según las que hay para cada señal de entrada de prueba. También pueden dividirse en cuanto a características para el régimen permanente y el transitorio, etc. En definitiva, se puede hacer una red conceptual usando esta relación “is-a” con el significado que se ha visto. Con la relación estructural expresada por medio de los slot, introducida previamente, tendríamos otra red conceptual. En general, con cada relación que introducimos en el modelo tenemos una red conceptual que nos relaciona unos conceptos con otros. Estas redes conceptuales son vitales a la hora de manejar el conocimiento y generar definiciones o respuestas a preguntas del usuario.

En cuanto a las representaciones gráficas, su modelado presenta dificultades extra ya que deben modelarse conceptos y relaciones pertenecientes al dominio de la geometría euclídea y el razonamiento gráfico. Es necesario modelar conceptos como punto, área, curva, y relaciones topológicas como la posición relativa, corte, inclusión, intersección, etc.

3.2.1 Modelado de tareas

Existen varios métodos para el diseño de compensadores en el lugar de las raíces. En el presente trabajo se ha estudiado el diseño de compensadores de adelanto-retraso de fase. En principio se parte de una serie de especificaciones de funcionamiento que pueden venir dadas como especificaciones para el régimen permanente y/o para régimen transitorio. Las especificaciones sobre régimen permanente suelen satisfacerse colocando una red de retraso de fase cerca del origen de coordenadas de forma que el lugar de las raíces no se vea modificado con esta acción, permitiendo de este modo separar el diseño para el régimen transitorio del de régimen permanente (se supone una estructura simple de realimentación unitaria con el compensador en serie con el proceso). En cuanto al régimen transitorio, el diseño suele resolverse hallando las posiciones de un polo y un cero reales correspondientes a una red de adelanto de fase y colocando una ganancia tal que los polos dominantes del sistema total estén en una posición que se corresponda con las especificaciones de diseño.

El diseño para régimen transitorio tiene por tanto mucho más contenido conceptual. Las especificaciones de régimen transitorio se traducen a una zona de diseño en el gráfico del lugar de las raíces, zona en la que deberán localizarse los polos dominantes del sistema total una vez calculados los parámetros del controlador. Seguidamente se comprueba si un ajuste de la ganancia (un controlador proporcional) puede llevar los polos dominantes a la zona de diseño. Para que esto ocurra alguna de las ramas del lugar de las raíces debe pasar por la zona de diseño, definiendo un rango de posibles valores de K . Para este rango de la ganancia ninguna rama del lugar de las raíces debe estar en el semiplano real positivo. Bajo estas condiciones se podría llevar a cabo el ajuste de la ganancia. Una vez fijada una ganancia, habría que ver si el diseño se comporta de acuerdo a las especificaciones simulando para ello el modelo total y comparando los resultados con los de las especificaciones. Si el resultado no es satisfactorio habría que iterar de nuevo, diseñando un nuevo valor de la ganancia o llegando a la conclusión de que el ajuste de la ganancia no va a servir para alcanzar las especificaciones.

Si el ajuste de la ganancia no es suficiente para alcanzar las especificaciones de diseño en régimen transitorio hay que diseñar una red de adelanto de fase para modificar la forma del lugar de las raíces para que éste pase por la zona de diseño. El diseño de la red de adelanto de fase consiste en hallar la posición de un polo y un cero, ajustando primero el cero y calculando analíticamente la posición del polo. Una vez que el lugar de las raíces pasa por el área de diseño hay que calcular el valor de la ganancia y volver a comprobar mediante simulación si el diseño final cumple las especificaciones. Si no las cumple habrá que hacer una nueva iteración de diseño buscando una nueva posición del cero por ejemplo.

El modelado de la tarea de diseño consiste en localizar las tareas y subtareas en las que aquéllas se pueden dividir y que en conjunto forman el proceso descrito anteriormente. Además será necesario expresar las relaciones existentes entre las diferentes tareas, relaciones que pueden ser de secuenciación (qué tareas van antes de cuales), de influencia mutua, etc. En general habrá una descripción de la secuencia de ejecución de tareas para llegar al objetivo final. Además cada tarea de uno u otro nivel de agregación tendrán una serie de precondiciones para poder ser llevadas a cabo y una serie de consecuencias o decisiones de diseño como consecuencia de su ejecución.

La principal característica distintiva de la tarea de diseño en control es que suele ser un procedimiento iterativo (es decir, hay tareas de rediseño). Muchas de las tareas del diseño en el lugar de las raíces son procedurales, es decir, tienen una forma de ser calculadas por métodos analíticos. Existen otras que por el contrario presentan un alto grado de contenido heurístico, entre las que se pueden citar a modo de ejemplo la determinación de un área de diseño a partir de las especificaciones iniciales o la colocación del cero del compensador de adelanto de fase. En estos casos la modelización requiere recoger este contenido heurístico y presenta retos más complejos que las tareas procedurales. Para modelar estas tareas se pueden utilizar aproximaciones similares provenientes de otros campos como el diseño paramétrico, razonamiento con incertidumbre, etc.

El modelo conceptual creado representa la tarea como una regla en la que las condiciones están formadas por los hechos que deben cumplirse para poder ejecutarse y las consecuencias son decisiones de diseño (valores para los parámetros del controlador o decisiones sobre la ejecución de otras tareas). Se ha dotado de una estructura multinivel para las reglas de forma que hay reglas “de bajo nivel” y reglas “de alto nivel” o metarreglas, que controlan la ejecución de otras reglas.

4 APLICACIÓN

Se ha creado una aplicación para comprobar la estructura de conocimiento. La aplicación es una especie de tutor inteligente que realiza el diseño de controladores y proporciona explicaciones sobre el proceso de diseño al usuario. Además, también ofrece descripciones sobre los diferentes conceptos a partir de la estructura del conocimiento interna. En la figura 1 puede verse una ventana de la aplicación.

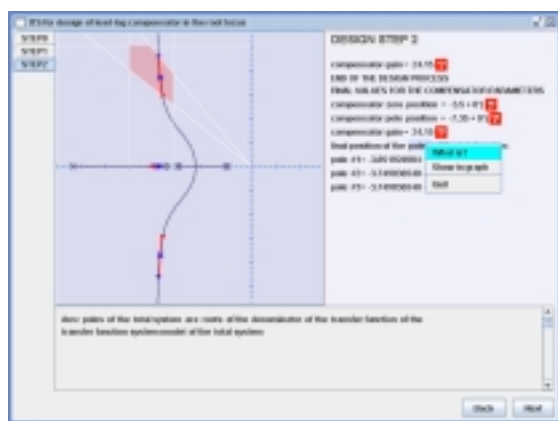


Figura 1: Pantalla de la aplicación

Cabe destacar que cada explicación que se ve en la ventana está creada a partir de la ontología y que cada concepto que aparece es una etiqueta activa sobre la que se puede hacer clic y acceder a un menú de posibles preguntas sobre ese concepto. Los elementos de la gráfica del lugar de las raíces también son activos e internamente están unidos a la estructura conceptual. Podemos tener una descripción generada automáticamente de todo el proceso de diseño, cada decisión de diseño tomada se presenta al usuario, ofreciéndole la oportunidad de preguntar por qué se ha realizado dicha decisión.

El conocimiento es parcial por el momento, por lo que el conjunto de funciones de transferencia a utilizar está acotado y no es libre. En un futuro se abordará la extensión a cualquier tipo de función de transferencia.

La aplicación se ha desarrollado utilizando Protégé [6] como herramienta editora de la ontología, Java como lenguaje de programación y aplicaciones de respaldo para cálculos numéricos como Matlab o Maple, cuyas interfaces con la ontología también han sido formalizadas en la misma. Conviene citar que, al contrario que en las técnicas de orientación a objetos, no se crean clases ni instancias a partir de los conceptos del modelo sino que la aplicación en sí son una serie de clases abstractas que “interpretan” la estructura de conocimiento plasmada en la ontología..

5 CONCLUSIONES

Se ha construido un modelo conceptual para las técnicas del lugar de las raíces y a partir del mismo se ha generado una aplicación que hace las veces de tutor inteligente en este campo. Se ha limitado el estudio al lugar de las raíces, lo supone perder parte del conocimiento que puede servir para hacer el diseño de forma más sencilla, pero lo que se intenta en esta primera aproximación es precisamente representar cómo el conocimiento del lugar de las raíces se corresponde con el comportamiento de un sistema hasta el punto en que esto se puede hacer razonando solamente en el plano complejo. Un futuro modelo conceptual incluirá también los métodos que usan la respuesta en frecuencia, representando también las relaciones entre estos métodos y el del lugar de las raíces y usando los dos conjuntamente para llegar a un diseño. Este hecho es especialmente importante y es una de las aplicaciones más provechosas que este tipo de aplicaciones puede tener.

Referencias

- [1] Gruber, T. (1993) "A translation approach to portable ontology specifications" *Knowledge Acquisition*, 5, pp. 199-220
- [2] Guzmán, J. L., Aström, K. J., Dormido, S., Häggglund, T., Pigué, Y. (2006), "Interactive learning modules for PID control" In 7th IFAC Symposium in Advances in Control Education, Madrid, Spain
- [3] Johanson, M., Gäfvert, M., Aström, K. J., (1998) "Interactive tools for education in automatic control" *IEEE Control Systems Magazine*, 18:3, pp. 33-40
- [4] Musen, M. A. , (1988) Modern Architectures for Intelligent Systems: Reusable Ontologies and Problem-Solving Methods. In C.G. Chute, Ed., AMIA Annual Symposium, Orlando, FL, pp. 46-52.
- [5] Palma, J.T., Paniagua, E., Martín, F., Marín, R., (2000) Ingeniería del Conocimiento. De la Extracción al Modelado de Conocimiento, Revista Iberoamericana de Inteligencia Artificial, Numero 11, Volumen 1, pp. 46-72
- [6] Protégé (2005) The Protégé Ontology Editor and Knowledge Acquisition Tool, Stanford Medical Informatics Laboratory, <http://protege.stanford.edu/>

MODELACION Y PREDICCION CON REDES PROBABILISTICAS

Caso de estudio: La Rugosidad Superficial

M. Correa

Instituto de Automática Industrial, CSIC.
Departamento de Informática Industrial. Crta. Campo Real Km. 0,200
La Poveda, 28500 Arganda del Rey, Madrid. España, macorrea@iai.csic.es

C. Bielza

Universidad Politécnica de Madrid, Departamento de Inteligencia Artificial
Facultad de Informática 28660, Boadilla del Monte, Madrid. España, mcbielza@fi.upm.es

M. de J. Ramírez

Instituto Tecnológico de Monterrey (ITESM) Av. Eugenio Garza Sada N°2501 sur
64849 Monterrey, Nuevo León, México, miguel.ramirez@itesm.mx

J. Pamies Teixeira

Faculdade de Ciências e Tecnologia-UNL
Quinta da Torre, 2829-516 Caparica (Portugal), jorge.pamies@fct.unl.pt

Resumen

En este trabajo se introducen las redes probabilísticas, concretamente las redes bayesianas para la predicción probabilística y el diseño de modelos conceptuales.

Se presenta un modelo para predicción de Rugosidad Superficial (Ra) desarrollado con redes bayesianas, basado en datos experimentales tomados en un centro de mecanizado a alta velocidad en un proceso de fresado. Se consiguieron resultados de precisión en la clasificación general de tipos de Ra de 88.5% y dependiendo del tipo de hasta 100%.

Palabras Clave: redes probabilísticas, redes bayesianas, rugosidad superficial (Ra), fresado a alta velocidad.

lineales están justificadas en que son muy fáciles de construir, particularmente los clasificadores, han tenido buen desempeño en otras aplicaciones de esta índole y son muy intuitivas [1].

Las Redes Probabilísticas están formadas por un conjunto de variables y su distribución de probabilidad conjunta. Esta es una herramienta capaz de obtener un modelo de dependencia entre todas y entre cada una de las variables disponibles. Este tipo de redes permite, además, cuantificar estas relaciones con la definición de una función de probabilidad conjunta de las mismas, como producto de funciones locales de probabilidad condicionada (la probabilidad de cada nodo condicionado a sus padres). En la ecuación 1 se muestra una función de probabilidad de Ra según una red bayesiana generada (Figura 1).

$$P(Ra, v, rpm, Va, Dh, fz, ap, ae, Cv, Hb, S) = P(Ra) \\ P(v|Ra, rpm) P(rpm|Ra, Va) P(Va|Ra, Cv) \\ P(Dh|Ra, rpm) P(fz|Ra, Va) P(ap|Ra, Va) P(ae|Ra, Va) \\ P(Cv|Va) P(Hb|Ra, Va) P(S|Ra, Va) \quad (1)$$

La función de probabilidad resultante se corresponde con las relaciones de dependencia e independencia codificadas en el grafo de la red, haciendo que estos modelos proporcionen una alternativa más potente y general que los métodos estándar basados en correlación.

2 REDES BAYESIANAS PARA PREDECIR Ra

Una Red Bayesiana es un grafo acíclico dirigido en el que cada nodo representa una variable y cada arco

1 INTRODUCCION

En la modelación de un sistema complejo, todos los datos y observaciones disponibles son susceptibles de ser utilizados. Uno de los paradigmas comunes de desarrollar modelos es a través de la estadística, sus metodologías basadas en diseño de experimentos y análisis de correlación. A pesar de obtenerse bastante rigor con un número más reducido de datos, los modelos estadísticos clásicos, como la regresión lineal, no han dado buenos resultados en la resolución de este tipo de problemas. Unas de las razones para adoptar los modelos basados en redes probabilísticas como solución a problemas no

una dependencia probabilística. Estas redes son una forma de representar el conocimiento bajo incertidumbre y dispone de métodos flexibles de razonamiento [3]. La figura No. 1 muestra un grafo de una red bayesiana con estructura de Naïve Bayes Aumentado a Árbol (TAN) como modelo de Rugosidad superficial (Ra), formada por 11 variables y las relaciones de dependencia que se establecen entre ellos. Este grafo determina las relaciones de dependencia que se crean entre las variables.

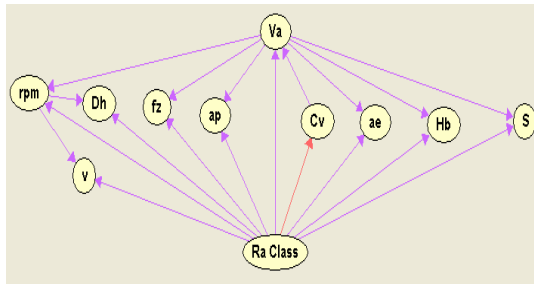


Figura 1: Ejemplo de red bayesiana con estructura TAN.

El algoritmo para construir un Tree Augmented Network (TAN) desarrollado por Friedman et al [6] consiste básicamente en una adaptación del algoritmo de Chow y Liu [4]. Para decidir los padres de cada nodo, tiene en cuenta la cantidad de información mutua condicionada a la variable clase, en lugar de la cantidad de información mutua en la que se basa el algoritmo de Chow-Liu. La cantidad de información mutua entre las variables discretas X e Y condicionada a la variable clase C se define en la ecuación No. 2 como:

$$I_p(X, Y | C) = \sum_i \sum_j \sum_r p(x_i, y_j, c_r) \log \frac{p(x_i, y_j, c_r)}{p(x_i, c_r) \cdot p(y_j, c_r)} \quad (2)$$

La construcción del árbol que forman las variables predictoras de un TAN consta de dos pasos. Dadas n variables, en el primer paso se calculan las cantidades de información mutua para cada par de variables (X, Y) condicionadas por la variable C .

Seguidamente debe construirse un grafo no dirigido completo con n nodos, uno por cada una de las variables predictoras, en el cual el peso de cada arista viene dado por la cantidad de información mutua entre las dos variables unidas por la arista condicionada a la variable clase.

En el segundo paso el algoritmo de Kruskal parte de los $n(n-1)/2$ pesos obtenidos en el paso anterior para construir el árbol expandido de máximo peso de la siguiente manera:

- Asignar las dos aristas de mayor peso al árbol a construir.

- Examinar la siguiente arista de mayor peso, y añadirla al árbol a no ser que forme un ciclo, en cuyo caso se descarta y se examina la siguiente arista de mayor peso.
- Repetir el paso 2 hasta que se hayan seleccionado $n-1$ aristas.

2.1 APRENDIZAJE

El desarrollo de sistemas basados en conocimiento motivó la investigación en el área de aprendizaje buscando la automatización del proceso de adquisición de conocimiento, uno de los principales problemas en la construcción de estos sistemas. Un aspecto importante del aprendizaje inductivo es obtener un modelo que represente el dominio de conocimiento, donde es importante obtener la información de dependencia entre las variables involucradas en el fenómeno en los sistemas en los cuales se busca predecir el comportamiento de algunas variables desconocidas a partir de otras conocidas. Las redes bayesianas son capaces de capturar esta dependencia entre las variables.

Dado un conjunto de datos que forman la base de conocimiento, se debe encontrar la estructura o grafo y los parámetros o probabilidades condicionadas, que mejor se ajusten a dichos datos utilizados como muestra. La estructura y los parámetros pueden ser estimados automáticamente a partir de los datos, con la ayuda de algoritmos eficientes, combinándolos en algunos casos con el conocimiento del experto.

En la construcción de la estructura de una red Bayesiana hay numerosas posibilidades, destacándose aquellas que tienen en cuenta dos aspectos principales:

- Una medida para evaluar la calidad del ajuste de cada estructura con respecto a los datos.
- Un método de búsqueda que genere diferentes estructuras hasta encontrar la óptima, de acuerdo a la medida seleccionada.

Hay varias formas de definir esa medida, siendo las más utilizadas la métrica Bayesiana (maximizando la probabilidad estimada de la estructura dados los datos) y la Longitud de Descripción Mínima (MDL), que estima la longitud (tamaño en bits) requerida para representar la probabilidad conjunta con cierta estructura. También hay varios métodos de búsqueda de estructuras.

2.2 MODELO PROPUESTO PARA LA PREDICCIÓN DE Ra

2.2.1 Concepto de Ra

En una pieza mecanizada Ra es el parámetro más representativo para la descripción de la textura superficial, condicionando muchos de los atributos y propiedades superficiales. Por eso ésta suele ser una de las variables de mayor importancia en el mecanizado a alta velocidad. La medición de Ra se hace con rugosímetros implicando inspecciones post-proceso usando métodos de control estadístico.

El término rugosidad superficial es cuantificado por parámetros relativos a características de la superficie. Es la variación vertical sobre una distancia de medida. La medida usada más comúnmente es Ra o rugosidad media, que es el área entre el perfil de rugosidad (Z_i) y su línea central dividido por la longitud de la evaluación por el número de evaluaciones(N).

$$R_a = \frac{1}{N} \sum_{i=1}^N |Z_i| \quad (1)$$

Según la norma ISO4288:1996 los valores están entre $0.006 \mu\text{m} < R_a < 80 \mu\text{m}$. Este parámetro es usado principalmente para supervisar el proceso de producción donde puede darse un cambio gradual del acabado superficial quizá originado por el desgaste de la herramienta de corte.

Como Ra es un promedio, los defectos en la superficie no tienen mucha influencia en sus resultados, significa que no es representativo en la detección de defectos dado que el valor de Ra no diferencia entre los picos y los valles.

Actualmente hay muchos grupos de investigación dirigidos al desarrollo de modelos de predicción de Ra [11][2][7][8][9][12]. Sin embargo no se ha alcanzado un consenso en el concepto general de la predicción y de los modelos a usar. Además, no existe en el mercado un dispositivo que permita medir en línea la rugosidad superficial.

En busca de este objetivo, este trabajo pretende contribuir en el desarrollo de un clasificador eficiente de Ra, usando redes bayesianas, que se profile como la clave para desarrollar un sensor virtual para la "medición" de Ra.

2.2.1 Hardware usado en la experimentación

Las pruebas se hicieron en un centro de mecanizado Kondia HS1000 equipado con un CNC abierto Siemens 840D. El material usado para las pruebas fue el aluminio 7075-T6 con HB145 y 5083-H111 con HB 65, con un perfil de dimensiones 170 x 100

mm. El diseño de la geometría elegida se puede apreciar en la figura 2, la profundidad de corte máxima fue de 10 mm. Se utilizaron herramientas marca Sandvik de 2 filos con 12 y 16 mm. de diámetro. Estas pruebas se realizaron con diferentes velocidades de corte y avance.

Para medir la rugosidad superficial se utilizó un rugosímetro digital Karl Zeiss modelo Surfcom 130. En el proceso de corte se midieron además las fuerzas de corte con un dinamómetro multi-componente con plato superior, y se instalaron acelerómetros en la pieza. Estas señales serán analizadas y utilizadas en trabajos posteriores continuación de este modelo inicial.



Figura 2: Diseño geométrico de las probetas.

2.2.3. Software

El clasificador fue desarrollado utilizando "Elvira" [13], programa desarrollado en Java destinado a la edición y evaluación de redes bayesianas y diagramas de influencia, que permite la generación de redes bayesianas a partir de un archivo de datos.

2.3 DESARROLLO DEL MODELO PARA PREDICCIÓN DE Ra

Como se comentó para la construcción del clasificador se aplicó el algoritmo TAN. Se tomaron once variables [5], algunas son parámetros de corte como: la profundidad y el espesor de corte (a_p y a_e), velocidad de corte (v), velocidad de avance (V_a), velocidad de giro (rpm), otras como las variables de la herramienta: diámetro de la herramienta (D_h) y avance por diente (f_z), propiedades del material como la dureza (HB) y algunas propias de la geometría de la pieza a mecanizar como el radio y forma (Cv), se tomaron probetas con 20 y 40 mm. de radio y formas cóncavas y convexas[10]. El diámetro fue dividido en 3 sectores, permitiendo hacer 3 réplicas de las mediciones para mejor representatividad estadística, o sea, minimizar el error experimental.

Las etiquetas de la variable Ra, se muestran en la tabla 1, y fueron asignadas de acuerdo al valor de rugosidad establecido en el estándar ISO 4287 e ISO 4288.

Tabla 1: Rangos de rugosidad del modelo.

Valor de rugosidad Ra		Código
Límites clase Ra(μm)	Promedio clase Ra(μm)	
0.5 – 1	0.63	Terso
0.3 – 0.5	0.4	Rugoso
0.1 – 0.3	0.16	Pulido

Para la construcción del modelo se han utilizado 97 datos, provenientes de la fase experimental.

Se selecciona la clasificación supervisada porque es la que se adapta al tipo de datos a usar. Los clasificadores son algoritmos dedicados "a interpretar" los datos provistos en el archivo de entrenamiento para generar la red bayesiana.

Partiendo de esta premisa, el objetivo de un modelo de clasificación consiste en clasificar correctamente casos nuevos, por tanto no debe hacerse la validación sobre los mismos datos con que se creó el clasificador. Por ello se tomó como método de validación el *K-Fold Cross-Validation* o validación cruzada.

Este consiste en dividir el conjunto total de casos en K subconjuntos disjuntos de aproximadamente el mismo tamaño. K-1 de dichos subconjuntos los utilizamos para entrenar el modelo, testándolo con el restante subconjunto. Este procedimiento se efectúa K veces. La estimación de la tasa de error en la clasificación se da como la media de las K tasas de error obtenidas. En nuestro caso la validación se efectuó con K=10.

Terminada la validación se genera una matriz de confusión, que consiste en una tabla de contingencia cruzando la variable derivada de la clasificación obtenida, con la variable que guarda la verdadera clasificación.

2.4 EVALUACION Y RESULTADOS

Inicialmente se construyó una red con la estructura Naïve Bayes para conocer cuál es la influencia de cada una de las variables con la clase de manera independiente.

En la figura 3 se muestran las distribuciones de probabilidad a priori calculadas a partir del conjunto de datos con el que se construyó la red.

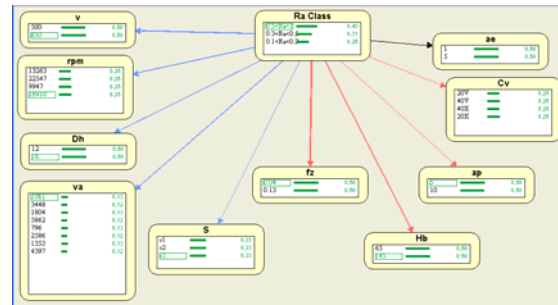


Figura 3: Distribución de probabilidades a priori – estructura Naïve Bayes

Los enlaces rojos muestran influencia positiva de la clase Ra con las variables fz, Hb y en menor proporción con las variables Cv y ap. Esto significa que a valores más altos de Ra la probabilidad de que avance por diente (fz) y dureza del material (HB) tomen mayores valores es más alta, presentándose la misma situación pero con menor proporción de aumento con la profundidad axial y la geometría de la pieza.

Los enlaces azules indican influencia negativa de la clase Ra sobre las variables v, rpm, Dh y Va. Significa que a mayores valores de Ra la probabilidad acumulada que los valores de velocidad de corte (v), velocidad de giro (rpm), diámetro de la herramienta (Dh) y velocidad de avance (va) tomen valores más altos disminuye. Igual situación pero en menor proporción se presenta con la variable sector (S).

Es importante tener en cuenta que además la estructura Naïve Bayes está basada en la hipótesis de independencia condicional de las variables predictoras dada la clase.

Un análisis realista acorde con el problema de predicción de Ra se obtiene con una estructura TAN. Esta es más indicada que la anterior porque no se toman las variables de forma independiente, sino que plantea la dependencia entre ellas y su relación con la clase. Para este caso se eliminaron las variables velocidad de avance (Va) y velocidad de giro (rpm) por ser redundantes. Estas pueden ser calculadas a partir de la velocidad de corte (vc) y el avance por diente (fz), y el sector (S) ya que no aporta información adicional a la red.

En la figura 4 se observa la red generada con este algoritmo. En este caso los enlaces muestran influencia indefinida en todas las variables, salvo fz que conserva la influencia positiva.

La influencia de la clase Ra está dada por los 3 tipos que puede adoptar, según los datos con que fue entrenada la red. La probabilidad más alta está en el

rango Terso ($0.5 < Ra < 1$) con probabilidad de 43% con influencia proporcional en todas las variables.

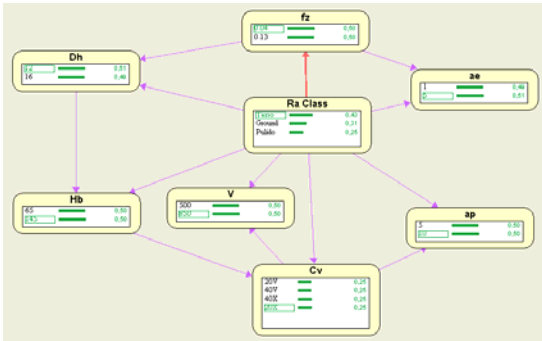


Figura 4: Distribución de probabilidades a priori – estructura TAN

Validando el modelo con el método de validación cruzada con $K=10$, se genera la matriz de confusión media de la figura 5, en donde se aprecia que el porcentaje de fiabilidad general del modelo es de 88.5%. En el tipo 0, es decir Pulido ($0.1 < Ra < 0.3$) es de 100%, el tipo 1 Ground ($0.3 < Ra < 0.5$) de 83.3% y tipo 2 Terso ($0.5 < Ra < 1$) de 75%. Esto significa que se estima que para cualquier nuevo dato introducido a la red perteneciente al tipo 0 será bien clasificado el 100% de las veces, del tipo 1 el 83.3% y del tipo 2 el 75%.

Precisión: 0.8854166666666666				
Matriz de confusión media				
	REAL	0	1	2
ASSIGNED	0	42	2	1
	1	0	25	5
	2	0	3	18

Figura 5: Matriz de confusión media

Introduciendo evidencia a la red, se asume que el valor de Ra esté en el rango Ground, en la figura 6 se observa como aumenta la probabilidad en las variables fz y V, expresa que si el valor de Ra está en el rango indicado hay más seguridad de que el valor de la probabilidad en el avance por diente y la velocidad de corte este en un valor estimado de 72%, además es más probable que el valor estimado de diámetro de la herramienta mayor este en 62% y en material blando en 71%. El comportamiento del avance por diente (fz) se ve reflejado en la figura 7.

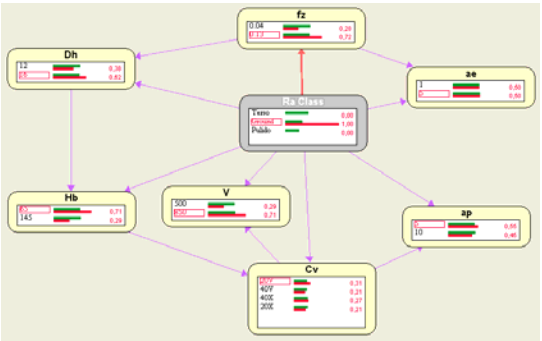


Figura 6: Distribución de probabilidades después de introducir como evidencia Ground ($0.3 < Ra < 0.5$).

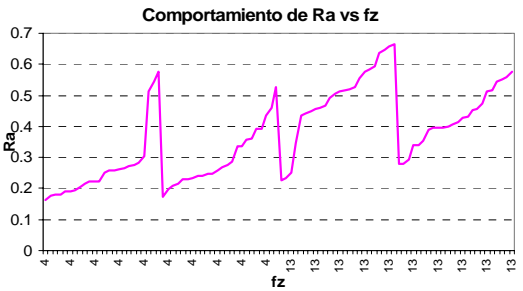


Figura 7: Comportamiento de Ra con aumento del avance por diente (fz)

De manera similar, figura 8, en el nodo avance por diente (fz) se introduce como evidencia el estado 0.13 y en el nodo velocidad de corte (V) el estado 500, se observa cómo aumenta el valor de la probabilidad de Ra del estado Pulido ($0.1 < Ra < 0.3$) de 25% a 56%. Esto significa que aumentando el avance por diente aunque se aumente la velocidad de corte es más probable conseguir mejores acabados superficiales. El estado Pulido ($0.1 < Ra < 0.3$) alcanza una probabilidad de 56% frente al 28% y 16% de los demás estados de Ra.

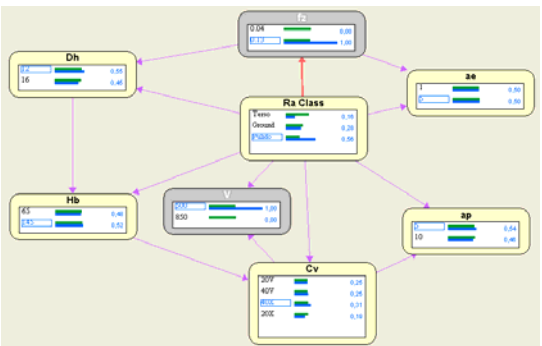


Figura 8: Distribución de probabilidades después de introducir como evidencia fz=0.13 y V=500

2.5 CONCLUSIONES Y TRABAJO FUTURO

Las redes probabilísticas y concretamente las redes bayesianas han demostrado ser una herramienta muy útil para la aportación de conocimiento en el desarrollo de un predictor de Ra.

El método presentado en este trabajo muestra el uso de las redes bayesianas con la estructura TAN para el desarrollo de un modelo empírico que es capaz de estimar el acabado superficial, concretamente Ra, con una fiabilidad de 88.5%, en procesos de mecanizado a alta velocidad.

El análisis de la matriz de confusión dio como resultados una exactitud de 100% en la clasificación del acabado en el tipo 0 correspondiente al rango $0.1 < Ra < 0.3$, en el tipo 1 de 70.9%, $0.3 < Ra < 0.5$ y en el tipo 2 de 79.1%, $0.5 < Ra < 1$.

Los resultados del modelo a partir de datos experimentales presentan una predicción estadísticamente satisfactoria.

Este trabajo es un avance en la búsqueda de un modelo para la predicción de Ra ya que se tomaron en cuenta variables como diámetro de la herramienta, dureza del material y geometría de la herramienta que no suelen tenerse en cuenta en el momento de modelar este tipo de procesos.

Para mejorar el desarrollo de los modelos se pretende utilizar señales adquiridas durante el proceso de corte como las fuerzas, aceleraciones y vibraciones.

Referencias

- [1] Acid, S., de Campos, L.M., Castellano, J.G (2005). "Learning bayesian network classifiers: searching in a Space of Partially directed acyclic graphs", *Machine Learning*, 59, pp. 213-235.
- [2] Azouzi, R., Guillot, M., (1997) "On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor fusion", *Int. J. Mach. Tools Manufact.* Vol. 37, No.93, pp. 1201-1217.
- [3] Castillo, E., Gutiérrez J.M., Hadi, A.S., (1997) *Sistemas Expertos y Modelos de Redes Probabilísticas*, Academia de Ingeniería, Madrid.
- [4] Chow, C.K., Liu, C.N., (1968) "Approximating discrete probability distributions with dependence trees", *IEEE Trans. on Information Theory*, 14, pp. 462-467.
- [5] Correa, M. Ramírez, M. de J., Alique, J.R., Rodríguez, C., (2004) Factores que afectan el acabado superficial en los procesos de mecanizado: técnicas de análisis y modelos, XXV Jornadas de Automática, Ciudad Real, España.
- [6] Friedman, N., Geiger, D., Goldszmidt, N., (1997) "Bayesian Network Classifiers", *Machine Learning*, 29, pp. 131-163.
- [7] Lou, M.J., Chen J.C., Li, C.M. (1999) "Surface roughness prediction technique for CNC end-milling", *Journal of Industrial Technology*, Vol. 15, No. 1.
- [8] Lou, S.J., Chen J.C. (1997) "In-process surface recognition of a CNC milling machine using the fuzzy nets method", *Computers ind. Engng.* Vol. 33, Nos. 1-2., pp. 401-404.
- [9] Lou, S.J., Chen J.C. (1999) "In-process surface roughness recognition (ISRR) system in end-milling operations", *International Journal of Advanced Manufacturing Technology*, pp. 200-209.
- [10] Ramírez, M. De J.; Correa, M.; Rodríguez, C.A.; Alique, J.R. (2005) "Surface Roughness Modelling based on Geometric Features and Process Parameters for High Speed Machining", *Proceedings of the IMECE'2005*, Orlando, Florida, USA.
- [11] Szalay, T., Alpek, F., Monostori, L., Markos, S., Viharios, Zs., (1996) "Investigation of machined surfaces using artificial intelligence methods", *Proceedings of the IXth International Conference on Cutting Tools*, pp. 635-640.
- [12] Tsai, Y.H., Chen, J.C., Lou, S.J. (1999) "An in-process recognition system based on neural networks in end-milling cutting operations", *International Journal of Machine Tools & Manufacture*, pp. 583-605.
- [13] The Elvira Consortium, (2002) "Elvira: An environment for creating and using probabilistic graphical models", *Proceedings of the First European Workshop on Probabilistic Graphical Models (PGM'02)*, pp. 1-11. Cuenca – Spain.

Modelado del Desgaste de Herramientas en el Mecanizado de Altas Prestaciones. Propuesta de Control Dimensional en Línea

José Vicente Abellán Nebot
abellan@esid.uji.es

Héctor Rafael Siller Carrillo
hector.siller@alumail.uji.es

Carlos Vila Pastor
vila@esid.uji.es

Fernando Romero Subirón
fromero@esid.uji.es

Depto. de Ingeniería de Sistemas Industriales y Diseño (ESID) de la Universitat Jaume I.
 Sos Baynat s/n C.P. 12071 Castellón, España.

Resumen

Este trabajo propone un modelo estadístico de predicción del desgaste de herramientas en operaciones de mecanizado de altas prestaciones. Se ha utilizado la sensorización de una máquina herramienta mediante la instalación de un dinamómetro piezoeléctrico y se ha establecido una relación lineal entre las fuerzas de corte medidas y el desgaste progresivo durante el tiempo de vida de la herramienta. Debido a que el desgaste tiene una influencia directa sobre la dimensión final de la pieza mecanizada, se plantea utilizar este modelo dentro de controladores externos a la máquina-herramienta que permitan calcular este error dimensional en proceso y compensarlo.

Palabras Clave: modelado, predictores, mecanizado de altas prestaciones, dinamómetro, desgaste de herramienta.

1 INTRODUCCIÓN

Para cualquier taller industrial que trabaje con procesos de arranque de viruta es de vital importancia mantener la calidad dimensional y geométrica de las piezas durante el mecanizado, y poseer cierta experiencia del proceso que le permita predecir el acabado superficial generado y el desgaste de la herramienta de corte para mantener una alta productividad y cumplir con los requerimientos del cliente. Sin embargo la complejidad del proceso de mecanizado dificulta en gran parte el desarrollo de aplicaciones o fórmulas matemáticas que faciliten esta tarea.

Uno de los parámetros más influyentes en la calidad dimensional de la pieza mecanizada viene dado por el propio desgaste de la herramienta de corte. Los desgastes modifican la geometría real del inserto de corte, modificando los espesores de material eliminado y generando variabilidad en las piezas procesadas finales. Debido al aumento en las exigencias de las calidades de las piezas en centros de mecanizado, es de interés plantear nuevas metodologías de compensación para minimizar el impacto de las variables que influyen en la calidad final de la pieza como profundidad del corte, sujeción de la pieza, vibraciones, etc. (Figura 1).



Figura 1: Variables que influyen en la calidad de las piezas mecanizadas.

En la actualidad la evaluación del estado de la herramienta de corte se realiza a través de la habilidad y la experiencia del operario, resultando un procedimiento caro y poco fiable. O bien se utilizan palpadores de reglaje de las herramientas, pero se pierde tiempo vital en el proceso o los dispositivos son costosos (sistemas láser por ejemplo). Si la herramienta de corte no se reemplaza al final de su vida útil se pueden generar defectos dimensionales en la pieza mecanizada, problemas de ensamblaje de

conjuntos de pieza, acabado superficial insatisfactorio e incluso daños irreparables en la misma máquina-herramienta. Si por el contrario la herramienta se cambia antes de que se alcance su vida útil, se tiene una subutilización de la herramienta de corte y una pérdida en el tiempo de producción [2].

El presente trabajo tiene como objetivo mostrar experimentalmente la posibilidad de modelar y predecir el desgaste de la herramienta de corte a través de la sensorización de la máquina-herramienta, concretamente mediante un dinamómetro piezoeléctrico. A su vez, se plantea la posibilidad de emplear esa predicción del desgaste de la herramienta de corte, para estimar la desviación de la calidad dimensional de la pieza a procesar, y permitir así una corrección de las dimensiones de las herramientas en las tablas de parámetros del controlador. Esta corrección permitirá minimizar las desviaciones geométricas sobre la pieza de trabajo y mantener así el proceso dentro de las tolerancias especificadas.

La estructura del artículo es como sigue. En la sección 2 se revisa la literatura más relevante sobre la monitorización del desgaste de la herramienta de corte. La sección 3 detalla el sistema experimental empleado. La sección 4 muestra la influencia del desgaste sobre la calidad dimensional de la pieza mecanizada. La sección 5 recoge los resultados experimentales mientras que en la sección 6 se realiza el modelado del desgaste de la herramienta. La sección 7 propone un esquema de control en línea y por último, la sección 8 concluye el artículo.

2 REVISIÓN DE LA LITERATURA

El desgaste en la herramienta de corte hace referencia al desgaste del filo de corte del inserto durante el proceso de mecanizado. En la literatura existe una gran cantidad de trabajos acerca de la predicción y monitorización del estado de la herramienta de corte [4],[6],[7].

En [5] se hace uso de la señal captada por dinamómetros, acelerómetros y sensores de emisión acústica, para monitorizar, en los dominios de tiempo y frecuencia, los cambios que pudieran existir debidos al mecanizado con herramientas desgastadas. En este trabajo se aprecia que, al desgastarse la herramienta, la amplitud de las vibraciones aumenta y aparecen nuevos armónicos en la etapa de máximo deterioro. En [8] se plantea un sistema informático de diagnóstico para el mecanizado de piezas complejas. El sistema genera mapas espaciales de fuerzas de corte, medidas en línea durante el mecanizado, que permiten realizar análisis posteriores causa-efecto de

posibles desviaciones geométricas ocasionadas por la condición de la herramienta.

El trabajo [10] presenta una estrategia para monitorizar la cantidad de desgaste de la herramienta y su relación con la variación de fuerzas de corte en operaciones de mecanizado. Se observa, en el dominio de frecuencias, que ciertos armónicos se incrementan con el desgaste mientras que otros permanecen sin cambio. Con esta información se ha desarrollado una simulación por ordenador del desgaste progresivo. El trabajo [9] compara el rendimiento de diferentes herramientas fabricadas con materiales y recubrimientos de última generación, en términos de variaciones de las fuerzas de corte en el transcurso de su vida útil. En este trabajo, para herramientas fabricadas de carburo de tungsteno con recubrimiento de TiAlN (nitruro de titanio aluminio) se observa una relación lineal entre desgaste y fuerzas de corte.

El trabajo [3] desarrolló un sistema multi-sensor (dinamómetro y acelerómetro) basado en la medición de fuerzas en X e Y y de vibraciones en el husillo. Se enfatiza el hecho de que el aplicar directamente la medición de los sensores para el modelado del proceso puede ser una tarea ineficiente, y que es necesario pre-procesar la información sensorial para emplear únicamente la parte de la medición que aporta información sobre el sistema. Por ello, para cada medición se obtuvieron 3 índices característicos de la señal de fuerzas, como son: la media, la varianza, y la frecuencia. En todos estos índices se encontraron relaciones directamente proporcionales con el desgaste de la herramienta. El trabajo [1] implementa un sistema multi-sensorial compuesto de acelerómetros y de la medición de la potencia consumida por la máquina. Se plantean diferentes estrategias de modelado, obteniéndose modelos de predicción tanto para el acabado superficial como para el desgaste de la herramienta de corte.

En la mayor parte de los trabajos enfocados a la predicción del desgaste de la herramienta de corte, se coincide en afirmar que son las fuerzas de corte las variables que mayor relación presentan con el estado de la herramienta de corte. Estas predicciones en línea, permiten efectuar acciones correctivas sobre la máquina de control numérico y mejorar así el acabado superficial y dimensional de las piezas mecanizadas.

3 SISTEMA EXPERIMENTAL

Con el objetivo de poner en marcha posibles sistemas de predicción, se propone la sensorización de la máquina-herramienta a través del uso de un dinamómetro piezoeléctrico de 8 canales. La

experimentación requerida para el modelado se centra en una de las operaciones de mecanizado más habituales en la industria, como es el planeado. Todo el equipo empleado en la experimentación, su montaje y el posterior tratamiento de las señales capturadas se muestra a continuación.

3.1 EQUIPO DE LABORATORIO

3.1.1 Máquina-herramienta

La máquina herramienta con la que se cuenta para el desarrollo experimental, es la mostrada en la figura 2.



Figura 2: Máquina-herramienta utilizada en el desarrollo experimental.

Las características principales de la máquina se muestran en la tabla 1.

Tabla 1: Características de máquina-herramienta

Centro de Mecanizado Vertical de 3 Ejes	
Marca/Modelo	Deckel Maho/DMC 70V
Potencia	40 KW
Vel. husillo máx.	15.000 rpm
Vel. avance máx.	30.000 mm/min
Área de trabajo	700x550x500 mm
Interpolación	Lineal, circular
Controlador	Heidenhain 426 PB
Accesorios	Palpador inalámbrico de piezas y sistema de reglaje de herramientas

3.1.2. Herramienta de corte

Para los experimentos se ha utilizado una herramienta de corte con plaquitas circulares, tal y como se muestra en la figura 3 y 4.

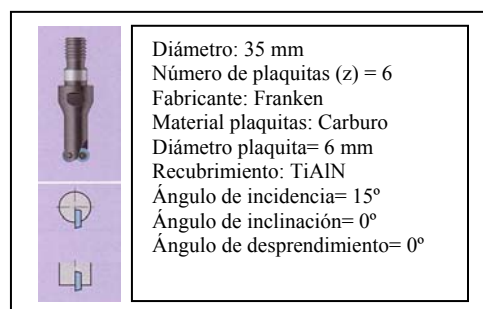


Figura 3: Herramienta utilizada.,

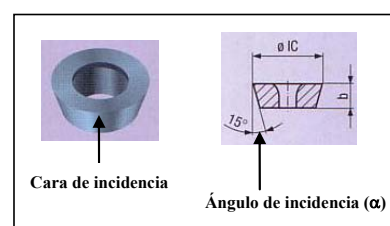


Figura 4: Plaquita de herramienta de corte.

3.1.3 Pieza a mecanizar

El tipo de material a mecanizar, en este caso es acero DIN 1.2080 con una dureza promedio de 60 HRC, utilizado en la fabricación de moldes y matrices, de dimensiones 250x250mm.

3.1.4 Dinamómetro piezoeléctrico

Los dinamómetros consisten en plataformas que incorporan células piezo-eléctricas que miden las fuerzas en las tres direcciones de operación de la máquina herramienta (x, y, z). La señal captada por dichos sensores es enviada a amplificadores que la condicionan para ser posteriormene capturada mediante sistemas de adquisición de datos (ver figura 5).

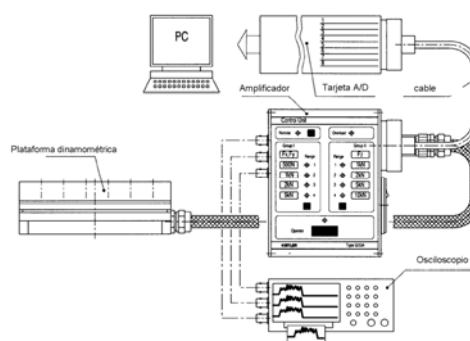


Figura 5: Ejemplo de un sistema de adquisición de fuerzas de mecanizado. Fuente: www.kistler.com

El dinamómetro empleado en la experimentación (Figura 6) consiste en un plato multicomponente de alta frecuencia natural, que provee mediciones dinámicas y cuasi-estáticas de los 3 componentes ortogonales de las fuerzas de corte (F_x , F_y y F_z) actuando en cualquier dirección en la parte superior del plato. La fuerza a ser medida es captada por tres pares de sensores de cuarzo sensibles a la presión en Z y a esfuerzos cortantes en X y en Y. Las señales son enviadas a amplificadores que convierten la carga eléctrica, previo ajuste de sensibilidad y escala, en unidades mecánicas. La salida de los amplificadores es posteriormente capturada por una tarjeta de adquisición de datos.

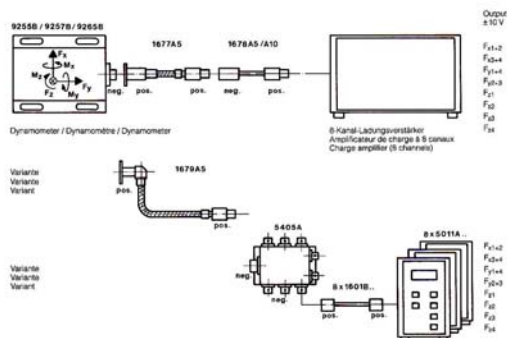


Figura 6: Sistema multi-componente dinamómetro-amplificadores de señal.

3.2 MONTAJE DEL EXPERIMENTO

La probeta a mecanizar se monta encima de la mesa dinamométrica (Figura 7) y se amarra por medio de tornillos, previa preparación de la probeta mediante taladrados pasantes. El dinamómetro se fija en la mesa de la máquina-herramienta mediante varias cuñas y bridas.



Figura 7: Montaje experimental.

3.3 TRATAMIENTO DE LAS SEÑALES

La captura de las señales se realiza mediante la tarjeta de adquisición de datos IOTECH 112 y el software DAQVIEW (Figura 8). Las señales capturadas a una frecuencia de 2000 muestras/s corresponden a las componentes en X, Y y Z de las

fuerzas de corte. Tras el mecanizado, los ficheros se analizan en MATLAB, obteniendo durante un tramo de corte de 1 segundo de duración el valor cuadrático medio de cada componente de la fuerza de corte (F_{x_RMS}), a través de la ecuación 1. El valor final empleado para el modelado corresponde a la fuerza resultante de corte, (F_{resul_RMS}) que se obtiene por la ecuación 2.

$$F_{x_RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N F_{x_i}^2} \quad (1)$$

$$F_{resul_RMS} = \sqrt{F_{x_RMS}^2 + F_{y_RMS}^2 + F_{z_RMS}^2} \quad (2)$$

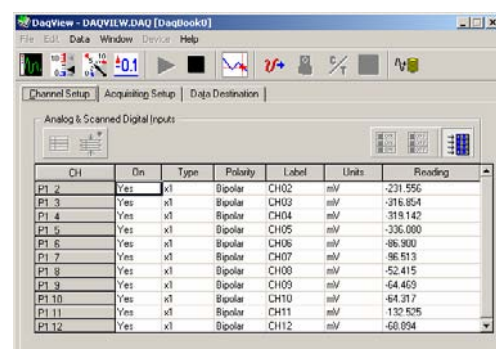


Figura 8: Software DAQVIEW para registro de las señales del dinamómetro.

3.4 DISEÑO DE EXPERIMENTOS

Para modelar el proceso de desgaste, se realizó el diseño de experimentos de la tabla 2. Como se observa los parámetros de corte establecidos fueron tres velocidades de corte (V_c): 60 m/min, 87,5 m/min y 100 m/min; y dos valores del avance por diente (S_z): 0,104 mm y 0,05 mm. Los valores de la profundidad axial (a_a) y profundidad radial (a_r) se mantuvieron constantes en todos los experimentos, y fueron de 0,2 mm y 21 mm respectivamente.

Tabla 2: Diseño de experimentos.

V_c (m/min)	S_z (mm/diente)	a_a (mm)	a_r (mm)
60	0,104	0,2	21
87,5	0,104	0,2	21
100	0,104	0,2	21
60	0,05	0,2	21
87,5	0,05	0,2	21
100	0,05	0,2	21

3.5 MEDICIÓN DEL DEGASTE DE LA HERRAMIENTA

Para la medición del desgaste se ha utilizado una lupa a 20 aumentos con cámara digital integrada, y las plaquitas han sido fotografiadas al finalizar cada trayectoria de planeado de la pieza de trabajo. El desgaste de las plaquitas de la herramienta ha sido medido de acuerdo a las directrices contenidas en el estándar ISO 4688-1. Bajo este estándar se especifica que para un desgaste localizado sobre la cara de incidencia de las plaquitas de la herramienta, se debe tomar como criterio de vida última un desgaste en la cara de incidencia de la herramienta (V_B) de 800 μm .

4 INFLUENCIA DEL DEGASTE DE LA HERRAMIENTA DE CORTE EN LA CALIDAD DIMENSIONAL DE LA PIEZA MECANIZADA

El desgaste de la herramienta de corte afecta a la dimensión superficial de la pieza mecanizada, de acuerdo a la relación mostrada en la figura 9.

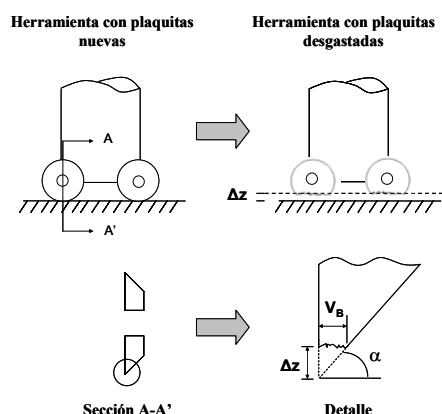


Figura 9: Relación entre desgaste de la cara de incidencia y altura de la plaquita.

Matemáticamente, la desviación en la dimensión en la dirección Z que no se mecanizaría debido al desgaste de la herramienta se puede modelar por la ecuación (3).

$$\Delta z = V_B \tan(\alpha) \quad (3)$$

Donde Δz es el decremento de altura en el eje axial de la herramienta debido al desgaste en la cara de incidencia (V_B) y α es el ángulo de incidencia.

En la experimentación de este trabajo, se ha controlado el Δz antes de cada pasada de mecanizado, con la ayuda de un palpador de piezas montado en el husillo de la máquina-herramienta. Con esta medida se asegura un constante arranque de material y una simulación del control en línea.

5 RESULTADOS EXPERIMENTALES

De acuerdo al diseño de experimentos, se han obtenido los resultados que se muestran en la tabla 3. La figura 10 muestra la evolución del desgaste de la herramienta de corte para una velocidad de corte (V_c) de 100 m/min y un avance por diente (S_z). Para todos los casos, en las fuerzas de corte resultantes se observan unas altas correlaciones lineales (R^2) con respecto al tiempo de mecanizado. Las figuras 11 y 12 muestran este hecho para $V_c=60$ m/min, $S_z=0,104$ mm y para $V_c=100$ m/min, $S_z=0,05$ mm respectivamente.

Tabla 3: Resultados Experimentales.

Variables				
V_c m/min	S_z mm	$F_{\text{resul_RMS}}$ N	V_B μm	Tiempo mins.
60	0,104	128	162,5	11,26
60	0,104	127,43	475	22,56
60	0,104	140	775	33,83
87,5	0,104	111	200	7,71
87,5	0,104	125	300	15,5
87,5	0,104	147	800	23,25
100	0,104	97,56	175	6,86
100	0,104	127,89	775	13,71
60	0,05	84,67	225	23
60	0,05	96,17	350	46
60	0,05	95	600	69
60	0,05	129,14	675	72,5
87,5	0,05	70	225	16
87,5	0,05	100	350	32
87,5	0,05	110	675	48
87,5	0,05	123	775	58
100	0,05	66	175	14
100	0,05	74	300	28
100	0,05	95	475	42
100	0,05	105	650	56

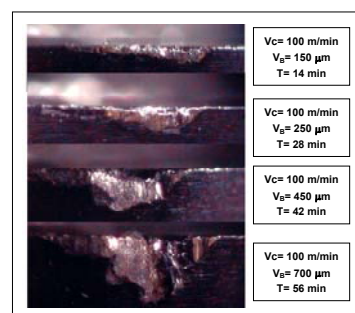


Figura 10: Fotografías del desgaste progresivo de la cara de incidencia para un experimento.

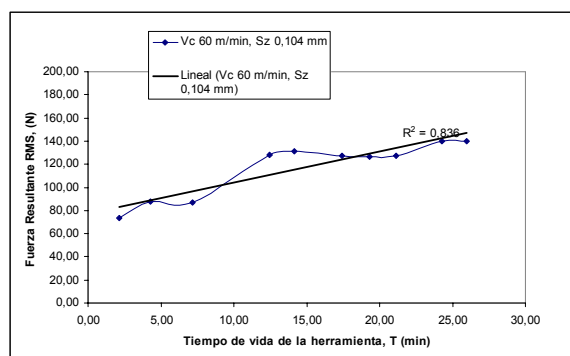


Figura 11: Gráfica de tiempo de vida de la herramienta vs. Fuerza Resultante para $V_c=60$ m/min y $S_z=0,104$ mm.

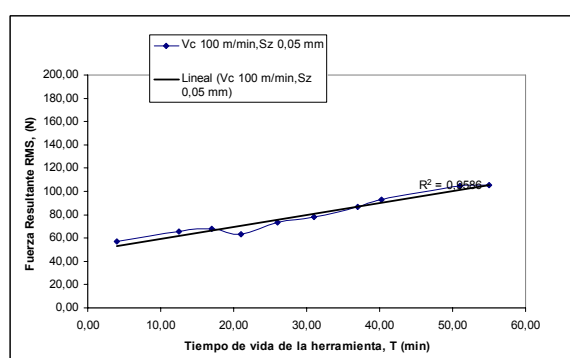


Figura 12: Gráfica de tiempo de vida de la herramienta vs. Fuerza Resultante para $V_c=100$ m/min y $S_z=0,05$ mm.

6 MODELO EXPERIMENTAL DEL DESGASTE DE HERRAMIENTA

Para predecir el desgaste de la herramienta de corte, se han empleado técnicas de modelado basadas en regresión múltiple. En principio se consideran en la regresión múltiple todas las interacciones de hasta orden 2 de las variables a excepción de la variable tiempo, para evitar la obviedad del modelo. Se hicieron dos regresiones, una para cada avance por diente S_z .

Para un $S_z=0,05$ mm, se realiza el primer análisis estadístico que emplea como variables de regresión las variables independientes V_c , F_{resul_RMS} y la interacción $V_c \times F_{resul_RMS}$. El análisis estadístico muestra altos p-valores, lo que indica que las variables no son significativas (Tabla 4). Eliminando uno a uno los p-valores más altos (menos significativos para el modelo), se llega a la regresión lineal final (ecuación 4), donde la fuerza resultante explica el 76,2% de la variabilidad del desgaste de la herramienta de corte (Tabla 5).

Tabla 4: p-valores iniciales para $S_z=0,05$ mm

Variable Regresión	p-valor	Variable Regresión	p-valor
Constante	0,026	F_{resul_RMS}	0,013
V_c	0,198	$V_c \times F_{resul_RMS}$	0,256

$$V_B = -1389 + 15,1 F_{res_RMS} \quad (4)$$

Tabla 5: p-valores del modelo para $S_z=0,05$ mm

Variable Regresión	p-valor
Constante	0,011
F_{resul_RMS}	0,003
% de correlación = 76,2%	

A través de un procedimiento similar, se tiene la regresión final que modela el proceso para un avance por diente $S_z=0,104$ mm (ecuación 5). Las tablas 6 y 7 muestran los p-valores de la regresión inicial y final.

$$V_B = -444 + 9,41 F_{res_RMS} \quad (5)$$

Tabla 6: p-valores iniciales para $S_z=0,104$ mm

Variable Regresión	p-valor	Variable Regresión	p-valor
Constante	0,774	F_{resul_RMS}	0,554
V_c	0,790	$V_c \times F_{resul_RMS}$	0,617

Tabla 7: p-valores del modelo para $S_z=0,104$ mm

Variable Regresión	p-valor
Constante	0,016
F_{resul_RMS}	0,000
% de correlación = 75,9%	

Para tener un mejor orden de magnitud acerca del ajuste de ambos modelos, se incluyen dos gráficas donde se pueden apreciar (Figuras 13 y 14).

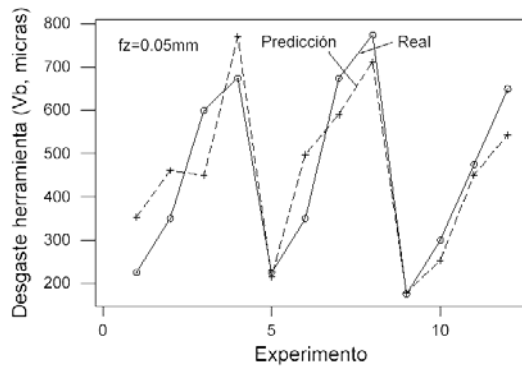


Figura 13: Comparativa entre valores experimentales y predicción. $S_z=0,05$ mm

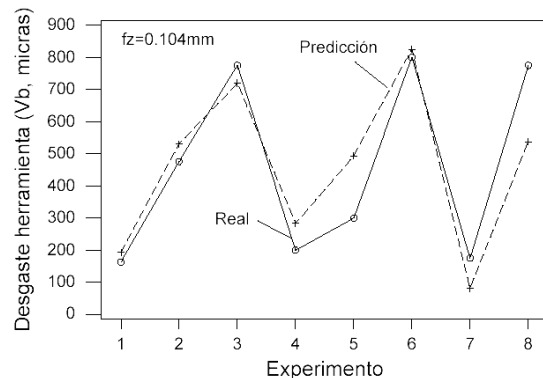


Figura 14: Comparativa entre valores experimentales y predicción. $S_z=0,104$ mm

7 PROPUESTA DE CONTROL

Tras la obtención de un modelo para la predicción aproximada del desgaste de la herramienta de corte, se puede hacer uso del modelo geométrico descrito en la sección 4 para predecir el posible desvío en el proceso de corte sobre la dimensión nominal deseada.

Por ello, es de interés plantear el desarrollo de un sistema de control/supervisión externo al control numérico de la máquina-herramienta, que permita predecir entre el procesado de cada pieza, el desgaste de la herramienta de corte a través de la adquisición de las fuerzas de corte. Esta predicción se basará en los modelos obtenidos off-line, y las predicciones serán función de los parámetros de corte. Con ello, se podrá compensar la desviación dimensional que se produciría en la pieza debido a ese desgaste, actuando sobre el control numérico antes del mecanizado. Este sistema viene reflejado en la figura 15.

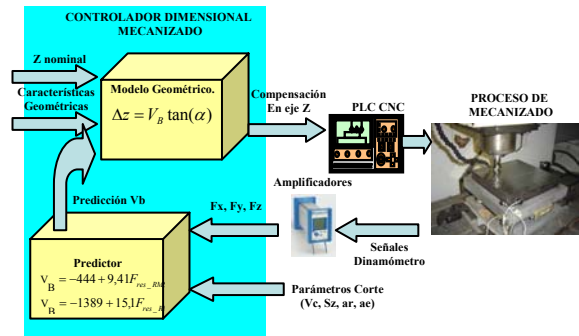


Figura 15: Sistema de control dimensional.

8 CONCLUSIONES

En este trabajo se presenta una de las problemáticas más habituales en los procesos de mecanizado de altas prestaciones, como es la predicción del desgaste de la herramienta de corte y su influencia sobre la calidad dimensional de las piezas mecanizadas. Por otro lado se propone un sistema basado en un dinamómetro piezoeléctrico para el modelado aproximado del desgaste de la herramienta de corte, y se valida este sistema mediante un conjunto de datos experimentales. Por último se plantea la posibilidad de desarrollar sistemas de control/supervisión a través de los cuales sea posible compensar los errores dimensionales debido al desgaste a través de la modificación de la trayectoria de la herramienta.

Agradecimientos

Este trabajo de investigación ha sido financiado por el proyecto "Integración de la Planificación, Ejecución y Control del Mecanizado a Altas Prestaciones en entornos de Ingeniería Colaborativa. Aplicación en la Fabricación de Componentes para Moldes de Azulejos Cerámicos" de la Universitat Jaume I. Participa en este proyecto la empresa MACER S.A., fabricante de moldes y matrices para la industria cerámica. Se agradece además el apoyo del Programa Alβan, Programa de becas de alto nivel de la Unión Europea para América Latina, beca nº E04D030982MX.

Referencias

- [1] Abellán, J., (2006) "Comparison of Modeling Approaches in Surface Roughness and Cutting", *CIRP. International Conference High Performance Cutting (HPC)*. Aceptado.
- [2] Bahr, B., (1997) "Sensor fusion for monitoring machine tool conditions", *International Journal of Computer Integrated Manufacturing*, v.10, pp. 314-323.

- [3] Chen, S., (2000) "Data fusion neural network for tool condition monitoring in CNC milling machining" *International Journal of Machine Tools and Manufacture*, v. 40, pp. 381-400.
- [4] Dimla, J. (1997) "Neural network solutions to the tool condition monitoring problem in metal cutting--A critical review of methods", *International Journal of Machine Tools and Manufacture*, v. 37, pp 1219-1241.
- [5] Haber, R., (2004) "An investigation of tool-wear monitoring in a high speed machining process", *Sensors and Actuators*, v. 116, pp. 539-545.
- [6] Jemielniak, K., (1995) "Tool and Process Monitoring -State of art and future prospects", *Scientific Papers of the Institute of Mechanical Engineering*.
- [7] Liang, S. (2004) "Machining Process Monitoring and Control: The State-of-the-Art", *Journal of Manufacturing Science and Engineering*, v. 126, pp. 297-310.
- [8] López Delacalle, L., (2006) "Recording of real cutting forces along the milling of complex parts", *Mechatronics*, v. 16, pp. 21-32.
- [9] Mativenga, A., (2005) "Wear and cutting forces in high speed machining of H13 using physical vapour deposition coated carbide tools", *Proceedings of the Institution of Mechanical Engineers*, pp. 191-199.
- [10] Sarhan, A., (2001) "Interrelationships between cutting force variation and tool wear in end-milling", *Journal of Materials Processing Technology*, v. 109, pp. 229-235.
- [11] Toh, C.K., (2006) "Cutter path strategies in high speed rough milling of hardened steel", *Materials & Design*, v. 27, pp. 107-114.

INSPECCIÓN AUTOMÁTICA DE CRISTALES EN FAROS DE VEHÍCULOS UTILIZANDO CONOCIMIENTO EXPERTO

Satorres Martínez, S., Gómez Ortega, J., López Paniza, J.M.
Grupo de Robótica, Automática y Visión por Computador
Escuela Politécnica Superior de Jaén. Universidad de Jaén.
Campus Las Lagunillas 23071 - Jaén
e-mail: {satorres,juango}@ujaen.es.

Resumen

En este artículo se propone una metodología para el posicionamiento óptimo de una cámara en la inspección automática de cristales de faros de vehículos. Partiendo del CAD del faro, distintos expertos marcarán zonas que deben ser inspeccionadas con un determinado nivel de detalle; con esta información, y teniendo en cuenta otro tipo de restricciones, se desarrollará un algoritmo genético para calcular las posiciones óptimas de la cámara.

Palabras Clave: Posicionamiento óptimo de una cámara, inspección automática, fuzzy, algoritmos genéticos, CAD.

1 INTRODUCCIÓN

La reducción del tiempo de fabricación y de inspección es un factor crucial en la obtención de mayores productividades. Este factor cobra una mayor importancia en industrias tan competitivas como la del automóvil.

Tradicionalmente, el control de calidad mediante visión por computador se ha centrado en el diseño de algoritmos encaminados a detectar distintos tipos de fallos, dejando a un lado el posicionamiento óptimo de la cámara. El objetivo es, por una parte, minimizar el número de imágenes a procesar (y de esta forma reducir el tiempo de inspección) y por otra, seleccionar las posiciones más adecuadas para inspeccionar un objeto.

Se puede plantear el problema del posicionamiento óptimo de una cámara con respecto a un objeto, como el procedimiento que define un conjunto mínimo de posiciones que satisfagan de forma eficiente y precisa una serie de requisitos de inspección. Estos requisitos pueden incluir variables como la resolución de la imagen (que afecta a la distancia a la que debe colocarse la cámara), la

orientación de la cámara respecto al objeto a inspeccionar, o el enfoque de la imagen.

Los distintos estudios que tratan el problema de posicionamiento óptimo de un sensor con respecto a un objeto [2], se pueden dividir en dos categorías en función de la forma que tienen de tratar el problema: *generar-probar* y *síntesis*. Las propuestas encuadradas en el primer grupo consideran la optimización de posiciones como un problema de búsqueda en un espacio restringido de soluciones. Generalmente, dividen el espacio esférico que rodea al objeto respecto al que se pretende posicionar el sensor, en un conjunto de posibles soluciones que son evaluadas imponiéndoles una serie de restricciones. Para conseguir un conjunto óptimo de posiciones en las que situar la cámara, se utilizan métodos heurísticos. Esto hace que sean fáciles de implementar pero tienen el inconveniente del alto coste computacional [9]. Dentro de este grupo son destacables los trabajos de S. Y. Chen, Y.F. Li [8], que utilizan algoritmos genéticos, y los de E. Dunn, G. Olague [6], que utilizan optimización multiobjetivo.

La otra forma de abordar el problema se conoce como *síntesis* y consiste en modelar las restricciones del problema como funciones analíticas; éstas han de cumplirse para las distintas posiciones en las que el sensor debe ser situado. Las soluciones obtenidas utilizando esta metodología son bastante precisas, pero las ecuaciones que modelan las restricciones no son fáciles de resolver para espacios de búsqueda con muchos grados de libertad [1] [3] [4].

Estudios recientes, como los realizados por Weihua Sheng et al. [10] [11], proponen una combinación de las metodologías anteriores para resolver el problema de posicionamiento óptimo. Con la información del CAD de la pieza a inspeccionar y con un modelo de la cámara, generan un conjunto óptimo de posiciones en las que situar este sensor. Además, también realizan una optimización en el recorrido de estas posiciones teniendo en cuenta las restricciones

cinemáticas del manipulador sobre el que se encuentra montada la cámara.

El trabajo que se presenta en este artículo se basa en una estrategia del grupo *generar-probar*, y particulariza el problema de inspección al caso del cristal de un faro de un vehículo. Este problema presenta una dificultad especial al ser el objeto a inspeccionar de un material transparente. Habitualmente, esta inspección se realiza hoy en día de forma manual, siendo un operario experimentado el que visualmente detecta los fallos en el cristal, que consisten en rayas de diferentes tamaños. También es común que haya zonas del cristal en las que, por ser visibles posteriormente, una vez el faro esté montado en el vehículo, exigen un nivel de detalle en la inspección mayor que otras zonas.

Este conocimiento experto se ha tenido en cuenta en el procedimiento planteado en este trabajo, a través de la definición de regiones del faro a las que se les asocia una función de pertenencia a un conjunto borroso que determina el nivel de detalle necesario en la inspección. Posteriormente, se utiliza un algoritmo genético para generar el conjunto de posiciones óptimas de la cámara.

El artículo se estructura de la siguiente forma: en la siguiente sección se presenta la descripción del problema que se pretende abordar. En la sección 3 se expone la forma en la que se tiene en cuenta el conocimiento experto del operario. La sección 4 muestra el procedimiento de obtención de las posiciones óptimas de la cámara, estando la sección 5 dedicada a mostrar los resultados obtenidos con la aplicación del procedimiento desarrollado en la sección 4. Finalmente, la sección 6 sintetiza las conclusiones del trabajo.

2 DESCRIPCIÓN DEL PROBLEMA

El problema consiste en determinar las posiciones óptimas en las que se tendría que situar una cámara para inspeccionar con un determinado nivel de detalle la zona vista del cristal del faro de un vehículo.

Los estándares de calidad son variables en función del cliente; es decir, cada cliente impone unas pautas de inspección determinadas, admitiendo fallos de hasta un cierto tamaño en las distintas zonas del cristal. Por regla general, se admiten un número determinado de defectos, de hasta 0.3mm como máximo, en la zona central del cristal. En los bordes suelen ser más permisivos llegando a tolerar fallos de más de 1mm (Fig. 1).

La inspección del cristal de un faro utilizando visión por computador es una tarea complicada, en la que la selección de la fuente de iluminación adecuada y la posición de la cámara respecto a esta fuente es fundamental para obtener un buen resultado.

Tanto la dificultad de inspección como el tamaño mínimo de defecto permitido no son constantes en toda la superficie del cristal. Aplicando lógica difusa, se combina esta información para conseguir el nivel de detalle con el que inspeccionar las distintas zonas.

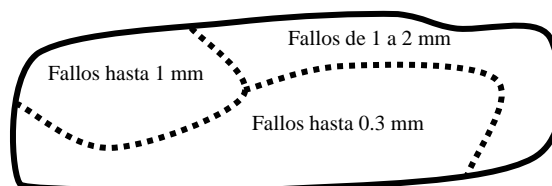


Figura 1. Croquis del cristal de un faro con los fallos permitidos en cada una de las zonas.

El nivel de detalle con el que se debe inspeccionar cada zona del cristal, determina la resolución exigida a la imagen utilizada para la inspección, lo que a su vez define el campo de visión necesario para conseguir esa resolución.

Con el nivel de detalle y teniendo en cuenta otras restricciones como la curvatura de la zona a inspeccionar, solapamiento entre los distintos campos de visión de las posiciones propuestas, etc., se implementa un algoritmo genético que indicará las posiciones óptimas de la cámara para inspeccionar el cristal con el nivel de detalle requerido (Fig.2).

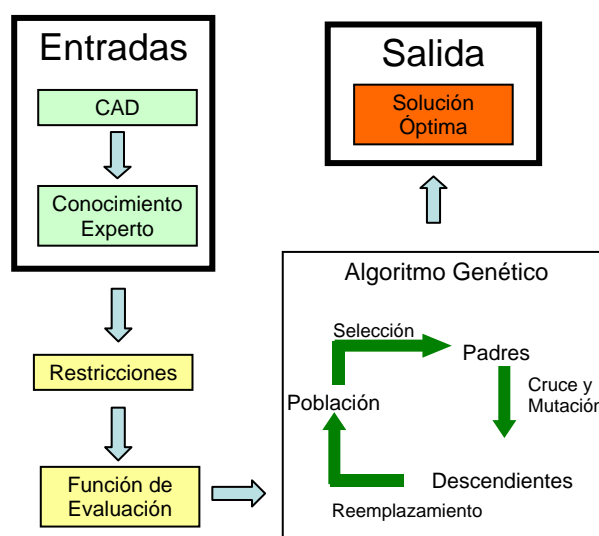


Figura 2. Diagrama de bloques del proceso.

3 EL CONOCIMIENTO EXPERTO APLICADO A LA OBTENCIÓN DEL NIVEL DE INSPECCIÓN

Para determinar el nivel de inspección necesario en cada zona de la pieza, se utilizará el conocimiento experto de diferentes características que será posteriormente combinado para determinar el nivel de detalle exigido en la inspección.

Basándose en la opinión de un experto, se determinan tanto las zonas de importancia estética subjetiva de la pieza como el grado de dificultad de la inspección. Cada zona se codifica usando un emborronamiento en función de la distancia a esas zonas. El algoritmo será:

Paso 1 – Se considerarán dos conjuntos borrosos: la importancia de una zona y su dificultad al ser inspeccionada. Un experto define, según su criterio, las zonas de la pieza que corresponden a ambas categorías: muy importante, importante y poco importante para la primera, y difícil, normal y fácil, para la segunda.

Paso 2 – Para introducir este conocimiento experto en el algoritmo es necesario codificarlo de forma matemática. Se emplea para ello un recubrimiento de cada una de las zonas definidas empleando esferas. Estas esferas serán las que realmente se utilicen en el algoritmo. El recubrimiento será válido si se cubre satisfactoriamente la zona asignada. En la figura 3 se representa a la izquierda la zona definida por el experto, en el centro se cubre la zona con esferas y a la derecha el recubrimiento.

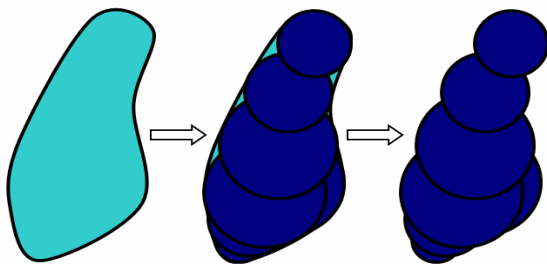


Figura 3. Recubrimiento satisfactorio.

Paso 3 – Dado que los datos de posición de la pieza son extraídos a partir del CAD, la información correspondiente a su geometría viene dada en sectores triangulares. Para cada sector de la superficie se evalúa su pertenencia a cada una de las zonas mediante una función de pertenencia que depende de la distancia del sector al centro de cada una de las esferas del recubrimiento. Si esta distancia es menor que el radio, la pertenencia al conjunto será total, mientras que si está comprendida entre el radio

y dos veces este radio pertenecerá de forma parcial. Esta función se representa en la figura 4 para una dimensión y en la figura 5 para dos dimensiones.

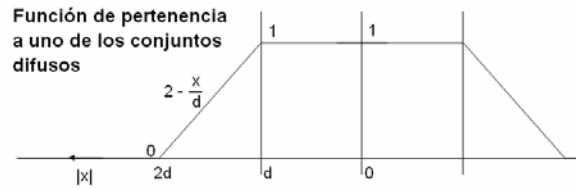


Figura 4. Función de pertenencia en una dimensión.

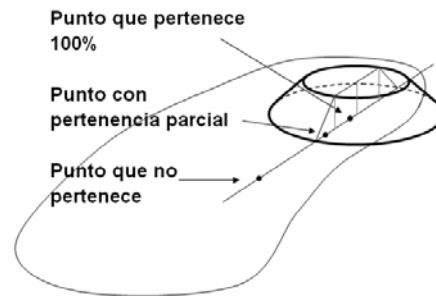


Figura 5. Función de pertenencia en dos dimensiones.

Las figuras 6 y 7 muestran un ejemplo de la aplicación de este algoritmo. Se observa cómo los diferentes sectores de la pieza son asignados tanto al nivel de importancia (Fig. 6) como a las zonas de dificultad (Fig. 7). En ambas figuras el color indica el grado de pertenencia a cada conjunto: en la figura 6 los rojos son poco importantes, los verdes importantes y los azules muy importantes, mientras que en la figura 7, los rojos son zonas fáciles, los verdes normales y los azules, zonas de gran dificultad de inspección.

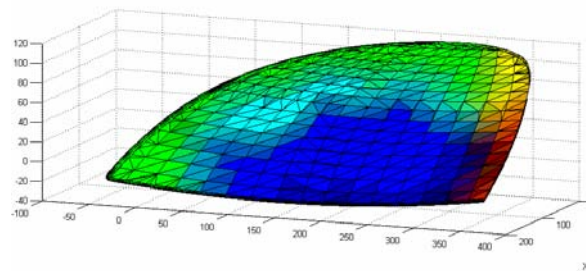


Figura 6. Niveles de importancia.

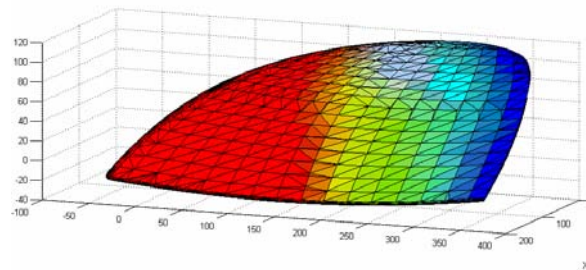


Figura 7. Niveles de dificultad.

Paso 4 – Se ejecuta, en base a los seis valores emborronados para cada sector de la superficie, el conjunto de las reglas difusas (Fig. 8 y Fig. 9).

Nivel de Inspección				
		Dificultad		
		Fácil	Normal	Difícil
Importancia	Muy importante	Alto	Alto	Alto
	Importante	Medio	Medio	Alto
	Poco Importante	Bajo	Bajo	Medio

Figura 8. Conjunto de reglas difusas.

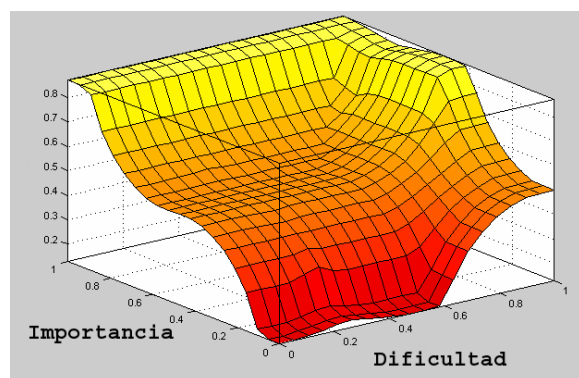


Figura 9. Representación de las reglas difusas.

Paso 5 – Se obtiene el nivel de inspección necesario para cada triángulo mediante el proceso de desemborronado. En la figura 10 se muestra el resultado de aplicar el algoritmo al ejemplo anterior. En ella los niveles altos de inspección (zonas brillantes) vienen determinados por importancias altas y son incrementados de cierta forma por la dificultad de cada zona.

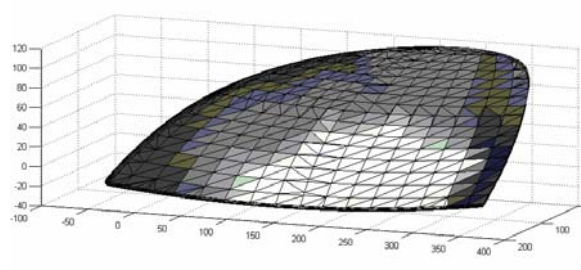


Figura 10. Resultado. Niveles de inspección.

4 OBTENCIÓN DEL CONJUNTO DE POSICIONES ÓPTIMAS DE LA CÁMARA

Se utiliza un algoritmo genético (AG) para determinar el conjunto óptimo de posiciones en las que colocar la cámara, e inspeccionar la zona vista del cristal del faro de un vehículo, con el nivel de detalle requerido por el cliente.

La aplicación de algoritmos genéticos es habitual en problemas de optimización en los que la solución debe cumplir una serie de restricciones; estas restricciones, en ciertos casos, pueden ser contradictorias.

El algoritmo genético utilizado es del tipo estacionario [5] con un reemplazamiento de la población inicial superior al habitual, que suele ser de un individuo en cada iteración.

El uso de AG en la optimización de un problema requiere la codificación de la población inicial, así como la formulación de la función de evaluación, (*fitness*, en terminología inglesa). Tanto una como otra deben ser particularizadas para el problema abordado.

En este caso se ha utilizado un cromosoma, representado en la figura 11, para codificar cada uno de los individuos de la población inicial. Este cromosoma se divide en 2 partes: genes de control y genes paramétricos [8]. Los genes de control son números binarios, en los que un cero desactiva los genes paramétricos que se corresponden con el gen de control mencionado y un uno activa los genes paramétricos correspondientes. Los genes paramétricos son números reales que representan, con respecto a un sistema de referencia, la posición de la cámara, orientación respecto al eje z y el tamaño del campo de visión $V_i = (x, y, z, \theta_z, l_1, l_2)$.

Siendo:

- . x_i, y_i, z_i : coordenadas geométricas en las que se debe situar la cámara.
- . θ_{zi} : ángulo respecto al eje z.
- . l_{1i}, l_{2i} : dimensiones del campo de visión de la cámara.

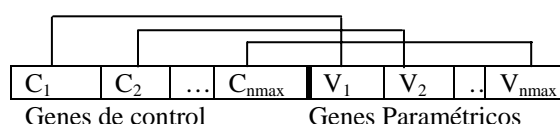


Figura 11. Estructura de un cromosoma de la población.

Como primera aproximación se ha considerado que la cámara se posiciona de forma perpendicular con respecto a la normal de la superficie que se va a inspeccionar. Por este motivo, no se experimenta rotación respecto de los ejes X e Y (Fig. 12).

El número de genes de control de la población inicial (n), es un número fijo que se ha estimado teniendo en cuenta el número máximo de imágenes necesarias para cubrir la totalidad del cristal del faro. El número de imágenes es inversamente proporcional al tamaño del campo de visión, de manera que el número de imágenes será máximo cuando el campo de visión sea mínimo.

Para que el algoritmo sea eficiente desde el punto de vista computacional, se ha estimado de forma heurística que el número máximo de genes de cada cromosoma sea de 168. Esto equivale a tener un total de 24 genes de control, que indican el número máximo de campos de visión.

El tamaño del cristal del faro es variable. En este caso, se ha trabajado con un cristal tipo de 450 x 150 mm de superficie. Al cubrir esta superficie con 24 posiciones, se obtiene un campo de visión de aproximadamente 75 x 56 mm.

Para estimar el campo de visión que se tendría de manera experimental, se ha utilizado el modelo óptico "Pin-hole" (Fig. 12).

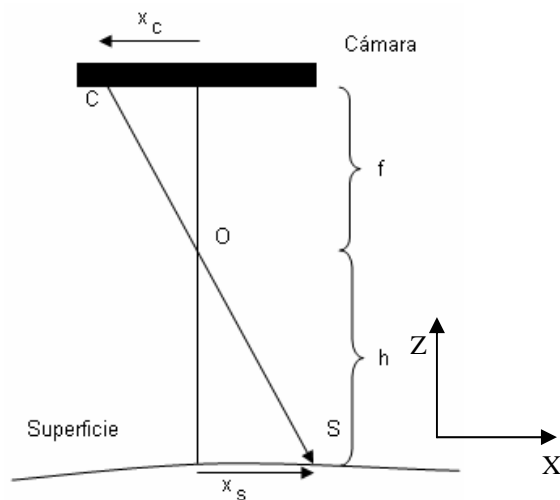


Figura 12. Modelo Pin-hole.

La relación entre el campo de visión de la cámara y el tamaño del CCD, se muestra en la ecuación 1 [7].

$$\frac{h}{f} = \frac{x_s}{x_c} \quad (1)$$

Siendo:

- . h: distancia entre la superficie a inspeccionar y la superficie de la lente.
- . f: distancia focal de la cámara.
- . x_s : tamaño del campo de visión.
- . x_c : tamaño del sensor CCD.

La cámara utilizada es una JAI CV-M77 de resolución 1024 x 768 y píxeles cuadrados de tamaño 4.76 μm , con una lente de 25 mm que, posicionada a una distancia de 400 mm, proporciona un campo de visión de unas dimensiones similares a las propuestas.

4.1 POBLACIÓN INICIAL

La población inicial está formada por 100 individuos, de los cuales el 25% se generan de forma manual, distribuyendo uniformemente las posiciones de la cámara sobre la superficie a inspeccionar para distintos niveles de detalle.

El 15% se generan distribuyendo uniformemente el campo de visión sobre la superficie a inspeccionar. En este caso, no obstante, las posiciones pueden variar un determinado rango entorno al valor en el que deberían posicionarse.

El resto de los individuos se generan de forma aleatoria, así como el número de posiciones activas en cada individuo (indicado por los genes de control), como su valor (indicado por los genes paramétricos). En el caso de ser gen paramétrico, este valor se genera de forma aleatoria entre ciertos límites dependiendo del tipo de parámetro que represente el propio gen.

4.2 FUNCIÓN DE EVALUACIÓN

La función de evaluación es la función a minimizar cuando se aplica el AG (Ec. 2). El resultado de esta ecuación se encuentra normalizado entre cero y uno.

Los miembros de esta ecuación están ponderados mediante pesos, lo que permite dar más importancia a unos términos de la ecuación frente a otros.

$$F_{EVALUACIÓN} = W_A \cdot \left(\sum_{i=1}^{i=pos.cam.} detalle_i + curvatura_i \right) + W_B \cdot N_P + W_C \cdot A_{NC} + W_D \cdot S_{min} \quad (2)$$

Donde:

- W_A , W_B , W_C , W_D . Son los pesos asignados a cada uno de los miembros de la función de evaluación.

- **Detalle.** Indica si el tamaño del campo de visión para la posición evaluada de la cámara, se corresponde con el que debería tener para el nivel de detalle requerido (Fig. 13).
- **Curvatura.** Indica el grado de curvatura de la zona del faro incluida en el campo de visión.
- N_p . Número de posiciones de la cámara para cada cromosoma de la población.
- A_{NC} . Área del cristal del faro no cubierta por el conjunto de los diferentes campos de visión de la cámara.
- S_{min} . Solapamiento entre los distintos campos de visión.

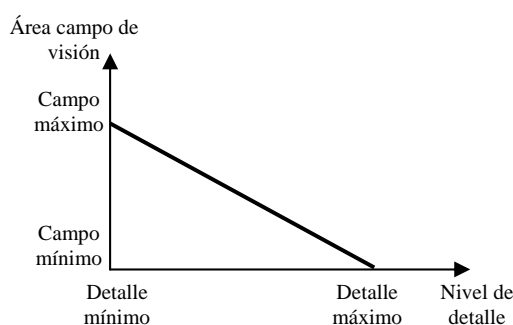


Figura 13. Relación entre el área del campo de visión y el nivel de detalle.

4.3 EVOLUCIÓN DEL ALGORITMO

En la evolución del AG se utilizan los parámetros y operaciones siguientes:

- **Longitud del cromosoma.** La longitud del cromosoma será $n + n \cdot k$, donde n es el máximo número de puntos de vista, que depende del tamaño del cristal a inspeccionar, de la configuración del sensor y del nivel de detalle requerido. El parámetro k es el número de variables independientes que definen cada posición.
- **Selección.** El método de selección será por torneo con tamaño dos.
- **Cruce.** La probabilidad de cruce es de 0.25, y se realiza el cruce por dos puntos.
- **Mutación.** La probabilidad de mutación es de 0.01 con respecto al gen. Los genes paramétricos y los de control mutan de la siguiente forma:
 - . *Genes de control:* mutación por cambio de bit.
 - . *Genes paramétricos:* mutación aplicando una función de distribución gaussiana

$g = g + \phi(\eta, \sigma)$ donde ϕ es la función de distribución gaussiana, η y σ son la media y la varianza respectivamente.

- **Reemplazamiento.** La nueva población estará formada por los hijos y los mejores individuos de la población anterior (cromosomas con un menor valor al aplicar la función de evaluación).
- **Condición de parada.** El algoritmo se da por finalizado cuando el indicador A_{NC} sea muy próximo a cero (toda el área de la superficie ha sido cubierta) y el solapamiento entre los campos de visión esté en un rango determinado. Si ninguna de las condiciones anteriores ha sido alcanzada, se finaliza en la iteración 1000.

5 RESULTADOS

Se presentan distintos resultados obtenidos con el AG para los niveles de detalle estimados mediante lógica difusa, partiendo de la información de distintos expertos.

En la figura 14 se representa el mejor valor al aplicar la función de evaluación, a cada uno de los cromosomas de la población para las 1000 iteraciones realizadas por el algoritmo.

Parte de la población inicial es dirigida a partir del conocimiento del campo de visión que puede tener la cámara y de las características geométricas de la pieza a inspeccionar. De esta forma, se optimiza el valor de la función de evaluación aplicada a la población inicial, asegurando una mayor rapidez en la convergencia al valor óptimo.

Finalmente, las posiciones obtenidas se muestran en la Fig. 15. Con 7 posiciones es posible inspeccionar la pieza por completo y con el nivel de detalle requerido.

Los pesos de la función de evaluación (Ec.2), utilizados para obtener los resultados de la figura 14, han sido los siguientes:

- $W_A: 0.4$. $W_B: 0.1$. $W_C: 0.4$. $W_D: 0.1$.

En este caso, se ha valorado más que los campos de visión obtenidos tengan el tamaño adecuado para inspeccionar las zonas con el nivel de detalle requerido y que se cubra el máximo posible de la superficie del cristal. Las zonas brillantes requieren una inspección con un nivel de detalle superior a las zonas oscuras. Por este motivo los campos de visión de estas zonas son de menor tamaño.

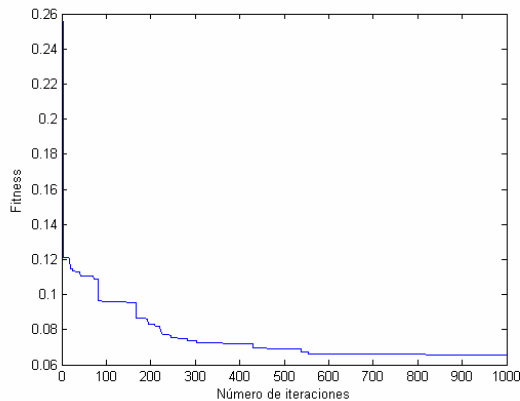
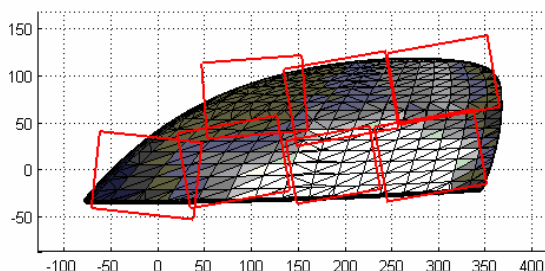
Figura 14. *Fitness/Iteraciones.*

Figura 15. Posiciones óptimas de la cámara.

6 CONCLUSIONES

Se ha desarrollado un algoritmo genético para determinar el conjunto óptimo de posiciones en las que debe situarse una cámara para inspeccionar el cristal del faro de un vehículo, con un nivel de detalle definido por un experto, en función de los requisitos del cliente al que va destinado el producto.

Modificando los coeficientes de ponderación de la función de evaluación (Ec.2), es posible obtener un conjunto de soluciones óptimas que satisfagan en distinta medida las restricciones planteadas.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el proyecto DPI2004-04458 del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2004/2007.

Referencias

[1] C.K. Cowan, P.D. Kovesi. "Automatic Sensor Placement from Vision Task Requirements".

IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 10, n° 3, May 1988.

- [2] K.A. Tarabanis, P.K. Allen, y R.Y. Tsai, "A survey of sensor planning in computer vision", IEEE Trans. Robot. Automat., vol. 11, pp. 86-104, Feb.1995.
- [3] K.A. Tarabanis, R.Y. Roger. "Computing Viewpoints that Satisfy Optical Constraints". Proc. CVPR 91: The Comput. Soc. Conf. on Comput. Vision and Patt. Recognition, 1991.
- [4] K.A. Tarabanis, R.Y.Tsai, y P.K. Allen. "The MVP Sensor Planning System for Robotic Vision Tasks". IEEE Transaction on Robotics and Automation, vol. 11, n° 1, Feb 1995.
- [5] D. Chafekar, J. Xuan, y K. Rasheed. "Constrained Multi-Objective Optimization Using Steady State Genetic Algorithms". Genetic and Evolutionary Computation 2003. Proceedings, Part. 1, pp.813-824, Springer.
- [6] E. Dunn, G. Olague. "Pareto Optimal Camera Placement for Automated Visual Inspection". IEEE/RSJ International Conference on Intelligent Robots and Systems. 2005.
- [7] R. Seulin, F. Merienne, y P. Gorria. "Dynamic lighting system for specular surface inspection". Proc. SPIE Vol. 4301, p. 199-206, Machine Vision Applications in Industrial Inspection IX. April 2001.
- [8] S.Y. Chen, Y.F. Li. "Automatic Sensor Placement for Model-Based Robot Vision". IEEE Transaction on Systems, man, and Cybernetics – Part B: Cybernetics, vol.34, n°1. Feb. 2004.
- [9] S.Y. Robert, M. Haralick, y L.G. Shapiro. "Automatic Sensor and Light Source Positioning for Machine Vision". Tech. Report EE-ISL-89-04, University of Washington, Seattle, WA, Sep. 1989.
- [10] W. Sheng, N. Xi, M. Song, y Y. Chen. "CAD-Guided Sensor Planning for Dimensional Inspection in Automotive Manufacturing". IEEE/ASME Transaction on Mechatronics, vol. 8 n°.3. Sep 2003.
- [11] W. Sheng, N. Xi, J. Tan, M. Song, y Y. Chen. "Minimum Viewpoint Planning for Dimensional Inspection of Sheet Metal Parts". Proceedings of the 2003 IEEE/ASME.

CONTROL VISUAL DEL ROBOT PARALELO ROBOTENIS

F. Sánchez¹, L. Ángel², L. Pari¹, J.M. Sebastián¹ y R. Carelli³

¹Dpto. de Automática, Ingeniería Electrónica e Informática Industrial, (Div. DISAM)
Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid UPM,
c/ José Gutiérrez Abascal 2, 28006-Madrid, España.

²Facultad de Ingeniería Electrónica, Universidad Pontificia Bolivariana
Km. 7 Vía de Piedecuesta, Bucaramanga, Colombia

³Instituto de Automática, Universidad Nacional de San Juan
Av. San Martín Oeste 1109, 5400 San Juan, Argentina

felixhuco2003@yahoo.es, langel@etsii.upm.es, lpari@etsii.upm.es, jsebas@etsii.upm.es, rcarelli@inaut.unsj.edu.ar

Resumen

El artículo describe el control visual de un robot paralelo, denominado Robotenis, con estructura de control abierta. El sistema ha sido diseñado y construido para realizar tareas en entornos dinámicos tridimensionales con objetos con un movimiento de hasta 2 m/s. La estrategia de control posee dos bucles: Uno interno basado en la información articular que actúa cada 0.5 ms, y otro externo basado en la información visual que actúa cada 8.3 ms.

Palabras clave: Control visual, Robot paralelo

1 INTRODUCCIÓN

Los sistemas de visión se utilizan cada vez con más frecuencia en las aplicaciones robóticas, aportando una información extremadamente útil sobre los objetos presentes en la escena, y permitiendo además conocer de una forma suficientemente precisa la posición y la orientación de los mismos. Sin embargo la integración de estos sistemas en tareas dinámicas presenta muchos aspectos no resueltos correctamente en la actualidad, y que son motivos de investigación en los principales centros de investigación ([10]). Entre ellos cabe destacar el diseño de estrategias de control visual de robots en tareas de seguimiento de objetos a alta velocidad (hasta 2 m/s), como las desarrolladas en la Universidad de Tokio ([8] y [11]).

Con el fin de poder estudiar e implementar distintas estrategias de control visual, el Grupo de Visión por Computador de la Universidad Politécnica de Madrid optó por el diseño y la construcción de la plataforma Robotenis, dentro del proyecto de investigación “Arquitecturas de teleoperación en entornos dinámicos modelables” (DPI 2001-3827-C02-01), financiado por el Ministerio de Ciencia y Tecnología. La Agencia Española de Cooperación Internacional (AECI) dentro del Programa de Cooperación Interuniversitaria de 2005, ha permitido igualmente la integración de investigadores extranjeros.

La plataforma llamada *RoboTennis* (ver Figura 1), es un robot paralelo de tres grados de libertad (está adaptado para incluir un cuarto grado adicional), con una estructura abierta para el control, que ha permitido la realización de tareas dinámicas a alta velocidad. En el presente artículo se describen las pruebas efectuadas en el seguimiento de un objeto (una pelota de ping-pong) que se mueve libremente con velocidad de hasta 1 m/s. Está prevista la realización de seguimientos a una mayor velocidad, y la realización de tareas que impliquen el golpeo de la pelota.

En términos generales, un robot paralelo consta de dos plataformas unidas por más de una cadena cinemática cerrada. Esta estructura presenta diversas ventajas con respecto a un robot serie, como son una mayor rigidez, precisión, capacidad de carga, velocidad y una menor inercia de los motores. La principal desventaja radica en la disminución del espacio de trabajo [1].

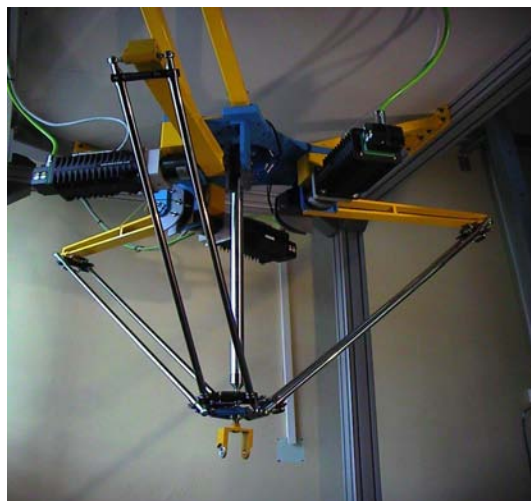


Figura 1. Sistema RoboTennis

La estructura mecánica del sistema RoboTennis está inspirada en el robot DELTA ([9]). El modelo cinemático, la matriz Jacobiana y el diseño óptimo

del sistema Robotenis han sido presentados en diversos trabajos previos ([3]). La estructura del robot ha sido optimizada tanto desde el punto de vista cinemático como dinámico. Así el método de diseño resuelve dos dificultades: La determinación de las dimensiones del robot y la selección de los actuadores. El análisis dinámico y el control articular han sido presentados en [2] y [4]. El modelo dinámico está basado en los multiplicadores de Lagrange e incorpora la influencia de las inercias de los antebrazos en el desarrollo de las estrategias de control. El sistema posee dos lazos de control. Uno interno basado en la información articular, que se ejecuta cada 0.5 ms, y que incorpora el modelo dinámico en un lazo de prealimentación y un PD en el lazo de realimentación. Otro externo, basado en la información visual, que se ejecuta cada 8.33 ms, y que se describe con más detalle en el presente trabajo.

El artículo se estructura de la siguiente forma: Tras la presente introducción, en el apartado segundo se describe brevemente el sistema Robotenis, ahondándose en el apartado tercero en el algoritmo de control visual propuesto e implementado. En el apartado cuarto se describen algunos de los resultados obtenidos, mientras que en el apartado quinto se resaltan las conclusiones obtenidas.

2 DESCRIPCIÓN DEL SISTEMA ROBOTENIS

En este apartado se describe el entorno de pruebas, los elementos que componen al sistema Robotenis y sus características funcionales. Más información se puede consultar en [1].

2.1 ENTORNO DE PRUEBAS

El objetivo definido para el sistema es que la pinza del robot sea capaz de seguir a una distancia de 600 mm a una pelota de ping pong sujeta con un hilo de la estructura fija (ver Figura 2) y que se mueva a una velocidad cercana a 1 m/s. A fin de facilitar la ejecución del control visual se ha optado por emplear una pelota de color negro sobre fondo blanco.

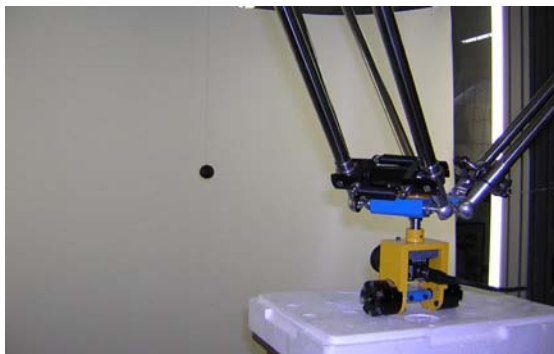


Figura 2. Entorno de trabajo

2.2 SISTEMA DE VISIÓN

El sistema Robotenis posee una cámara en el extremo del robot (ver Figura 3). La ubicación de la cámara combina la adquisición de un amplio campo visual, cuando el robot está alejado del objeto, con una adquisición más precisa cuando el robot está cerca del objeto. Este aspecto es muy importante para futuras aplicaciones del sistema, como puede ser golpear o coger una pelota. A destacar los siguientes aspectos:

2.2.1 Cámara

La cámara utilizada es la SONY XC-HR50. Sus principales características son:

- Captura de imágenes a alta velocidad, en 8.33 ms, con una resolución de 240 x 640 píxeles.
- Tiempo de integración de 1 ms.
- Escaneado progresivo.
- Reducido tamaño y peso muy ligero: 29 x 29 x 32 mm y 50 gr.
- Óptica de 6 mm

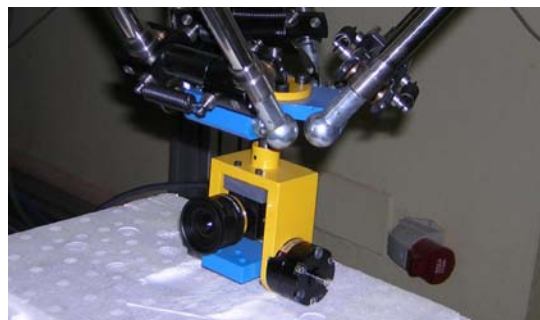


Figura 3. Cámara en el sistema RoboTennis

2.2.2 Tarjeta de adquisición

La tarjeta de adquisición utilizada ha sido la Matrox Meteor 2-MC/4. Permite la adquisición en modo doble buffer (adquisición simultánea con el procesamiento de la imagen previa), fundamental para ejecutar el bucle de control visual en 8.33 ms.

2.2.3 Procesamiento de imágenes

Una vez adquirida la imagen el sistema realiza la segmentación de la pelota sobre el fondo blanco. Como características visuales se calculan el centroide de la pelota y el diámetro de la misma, todas ellas con precisión subpixel. Mediante una calibración previa es posible determinar la posición espacial de la pelota. El algoritmo de control implica el conocimiento de la velocidad de la pelota, que se estima mediante un filtro de Kalman ([5] y [6]).

2.3 SISTEMA DE CONTROL

El hardware de control está formado por una tarjeta dSPACE 1103. En ella se ejecutan: la generación de trayectorias, el cálculo de los modelos cinemático y dinámico del manipulador, así como la implementación de los algoritmos de control. El sistema de accionamiento utilizado está compuesto por servomotores brushless AC (Unimotor), variadores de velocidad (Unidrive SP) y elementos reductores (más información en [1]).

2.4 CARACTERÍSTICAS FUNCIONALES

El control visual desarrollado está condicionado por las características propias de la aplicación y del sistema RoboTennis. Entre ellas cabe destacar las siguientes:

- Alta incertidumbre en los datos suministrados por el sistema de visión. El bajo período de muestreo (8,33 milisegundos) magnifica los errores en la estimación de la velocidad. Así por ejemplo con la pelota a 600 mm de distancia el diámetro de la pelota mide unos 20 píxeles. Si se comete un error de 0,25 píxeles en la estimación del diámetro, el error propagado en la estimación de la distancia será de 8 milímetros aproximadamente, y de 1 metro/segundo en la velocidad de la pelota, muy elevada para la necesaria continuidad del movimiento del robot. el filtro de Kalman disminuye parcialmente este problema.
- Continuidad de la velocidad del robot. El planificador diseñado para el sistema RoboTennis, necesita una cierta continuidad en la velocidad para evitar aceleraciones elevadas que penalizaría notoriamente el movimiento. Así un error de 8 milímetros con un período de muestreo de 8,33 milisegundos ocasionaría una aceleración de 12g, superior a las prestaciones del sistema. Es necesario destacar que el sistema RoboTennis es un sistema abierto, por lo que está previsto el cambio de referencia mientras se mueve, para aumentar sus prestaciones.
- El sistema RoboTennis posee, como cualquier sistema real, diversas limitaciones que deben de ser tenidas en cuenta en la estrategia de control para poder asegurar el éxito de la misma. Las dos principales son los retrasos y las saturaciones. Existe un retraso entre los datos suministrados por la cámara y los obtenidos de los sensores articulares. El retraso se estima en 2 periodos de muestreo (16.66 milisegundos), y es debido a la integración de la información luminosa, su transmisión y procesamiento. Igualmente es necesario resaltar que la velocidad del robot estará limitada por construcción (2.5 m/s), y deberá ser tenida en cuenta por la estrategia de control.

3 CONTROL VISUAL DEL SISTEMA ROBOTENIS

Los sistemas de coordenadas definidos se muestran en la Figura 4. Σ_w , Σ_e y Σ_c son los sistemas del mundo, del extremo del robot y de la cámara. ${}^c p_b$ es la posición relativa de la pelota con respecto al sistema de coordenada de la cámara, y ${}^w p_e$ es la posición relativa del extremo del robot con respecto al sistema de coordenadas del mundo, conocida a través de la cinemática directa. Se suponen conocidas y constantes las matrices ${}^w R_e$, ${}^w R_c$, ${}^e R_c$ y ${}^e T_c$, a través de las calibraciones del sistema.

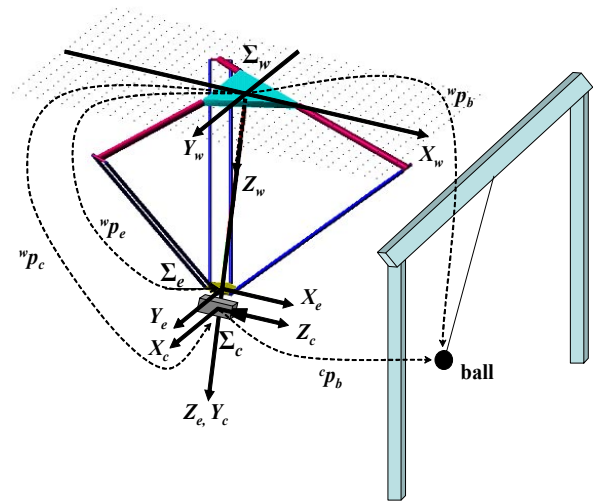


Figura 4. Sistemas de coordenadas involucradas

De entre las distintas alternativas existentes [7]), se ha elegido la utilización de un control basado en posición. Como se aprecia en el esquema básico de la Figura 5, la función de error se obtiene comparando la posición deseada del objeto (${}^c p_b^*(k)$) que en la presente aplicación se supone constante, con la posición obtenida por el sistema de visión (${}^c p_b(k)$). El controlador proporciona la velocidad deseada para el efector final. A través de un planificador en el espacio cartesiano y empleando la matriz Jacobiana del robot se obtiene la variación articular del robot. La variable k indica el instante de muestreo considerado.

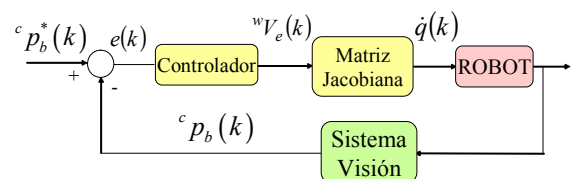


Figura 5 Control visual basado en posición

3.1 MODELO UTILIZADO

Según se aprecia en la Figura 5, el error se define como:

$$e(k) = {}^c p_b^* - {}^c p_b(k) \quad (1)$$

La posición de la pelota en el sistema de coordenadas de la cámara se puede expresar como:

$${}^c p_b(k) = {}^c R_w ({}^w p_b(k) - {}^w p_c(k)) \quad (2)$$

Por lo que sustituyendo se obtiene

$$e(k) = {}^c p_b^* - {}^c R_w ({}^w p_b(k) - {}^w p_c(k)) \quad (3)$$

Para asegurar que el error decrezca exponencialmente, supuesto el sistema estable, se elige

$$\dot{e}(k) = -\lambda e(k) \quad \text{con } \lambda > 0 \quad (4)$$

Derivando la ecuación (3), y teniendo en cuenta que ${}^c R_w$ es constante, se obtiene:

$$\dot{e}(k) = -{}^c R_w ({}^w v_b(k) - {}^w v_c(k)). \quad (5)$$

Sustituyendo las ecuaciones (3) y (5) en (4), se obtiene:

$${}^w v_c(k) = {}^w v_b(k) - \lambda {}^c R_w^T [{}^c p_b^* - {}^c p_b(k)] \quad (6)$$

Donde ${}^w v_c(k)$ y ${}^w v_b(k)$ representan las velocidades de la cámara y de la pelota respectivamente. Ya que ${}^w v_e(k) = {}^w v_c(k)$ la ley de control puede ser expresada como:

$${}^w v_e(k) = {}^w v_b(k) - \lambda {}^c R_w^T [{}^c p_b^* - {}^c p_b(k)]. \quad (7)$$

La ecuación (7) posee dos componentes: Una de predicción de la velocidad de la pelota (${}^w v_b(k)$) y otra de error de seguimiento ($[{}^c p_b^* - {}^c p_b(k)]$). Su ejecución implica un conocimiento exacto de todas las componentes, algo que no será posible. Una aproximación más realista es la siguiente:

$${}^w v_e(k) = {}^w \hat{v}_b(k) - \lambda {}^c R_w^T [{}^c p_b^* - {}^c \hat{p}_b(k)] \quad (8)$$

Donde las variables estimadas son utilizadas en lugar de las reales. La estructura básica de control visual sería la expresada en la Figura 6.

3.2 AJUSTE DEL PARÁMETRO λ

Un aspecto fundamental para el correcto funcionamiento del algoritmo de control visual es el ajuste del parámetro λ . En el presente trabajo se propone el cálculo de dicho algoritmo usando un algoritmo predictivo que intenta lograr el objetivo de control (${}^c p_b(k) = {}^c p_b^*$) en el menor número de muestras, teniendo en cuenta las limitaciones del sistema.

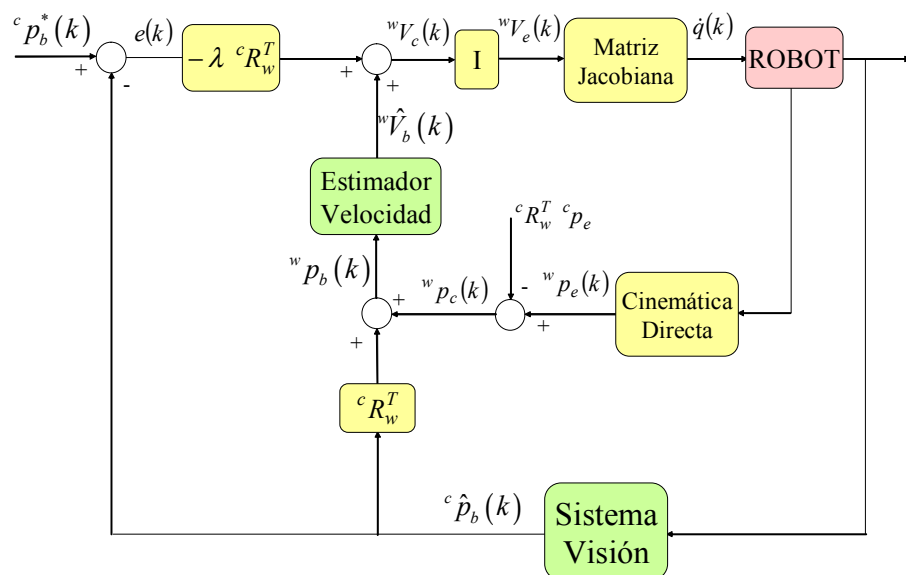


Figura 6. Arquitectura básica de control visual

La posición de la pelota en el sistema de coordenadas de la cámara se puede expresar en función de la posición de la pelota en el sistema de coordenadas del mundo y de la posición de la cámara en el sistema de coordenadas del mundo.

$${}^c p_b(k) = {}^c R_w [{}^w p_b(k) - {}^w p_c(k)] \quad (9)$$

La posición futura de la pelota en el sistema de coordenadas del mundo en el instante $k+n$ será:

$${}^w \hat{p}_b(k+n) = {}^w \hat{p}_b(k) + {}^w \hat{v}_b(k)Tn \quad (10)$$

Donde T es el período de muestreo (8.33 ms). Además la posición futura de la cámara en el sistema de coordenadas del mundo en el instante $k+n$ será:

$${}^w p_c(k+n) = {}^w p_c(k) + {}^w v_c(k)Tn \quad (11)$$

Si la ley de control pretende que se anule el error en el menor tiempo posible, sustituyendo la ecuación (9) para $k+n$, en la ecuación (2), se cumplirá:

$${}^c p_b^* - {}^c R_w ({}^w \hat{p}_b(k+n) - {}^w p_c(k+n)) = 0 \quad (12)$$

Sustituyendo (10) y (11) en (12), se obtiene:

$${}^c p_b^* = {}^c R_w [{}^w \hat{p}_b(k) + {}^w \hat{v}_b(k)Tn - {}^w p_c(k) - {}^w v_c(k)Tn] \quad (13)$$

Ya que ${}^w v_e(k) = {}^w v_c(k)$, la velocidad del extremo del robot se puede despejar, teniendo en cuenta la ecuación (2), obteniéndose la expresión:

$${}^w v_e(k) = {}^w \hat{v}_b(k) - \frac{1}{Tn} {}^c R_w^T [{}^c p_b^* - {}^c \hat{p}_b(k)]. \quad (14)$$

Comparando las ecuaciones (8) y (14), se define el parámetro λ como:

$$\lambda = \frac{1}{Tn}. \quad (15)$$

La ecuación (15) suministra una opción para ajustar el parámetro λ cumpliéndose la función objetivo en el menor número de muestras posible (n).

3.3 ALGORITMO IMPLEMENTADO

La arquitectura de control visual propuesta en la Figura 6 no considera las limitaciones físicas del sistema: Los retrasos y las posibles saturaciones motivadas por las velocidades máximas del robot. En la Figura 7, se representa un esquema de

control que incorpora y tiene en cuenta estas limitaciones. El término z^{-r} representa un retraso de r periodos para las señales de control. En el presente sistema se asume que la información suministrada por el sistema de visión (${}^c p_b(k)$), posee un retraso de $r=2$ periodos con respecto a la información articular suministrada por el sistema de control. Así la futura posición de la pelota se expresará como:

$${}^w \hat{p}_b(k+n) = {}^w \hat{p}_b(k-r) + {}^w \hat{v}_b(k-r)T(n+r) \quad (16)$$

Mientras que la futura posición de la cámara vendrá expresada por la ecuación (11).

La ecuación (15) permite ajustar el parámetro λ de la ley de control, teniendo en cuenta los siguientes aspectos:

- La ecuación (14) suministra la velocidad deseada del robot, que evidentemente no podrá sobrepasar la máxima velocidad permitida, lo que obligará a un ajuste del mínimo número de muestras en el que se puede cumplir la función objetivo. Este número será distinto para cada eje del robot, por lo que habrá que considerar el más restrictivo (el mayor), y se empleará en el cálculo del parámetro λ .
- A fin de lograr la mayor continuidad posible en la velocidad del robot, lo que facilita la labor del planificador de trayectorias que suministra la consigna al bucle interno basado en la información articular, se ajustará los parámetros λ de los ejes menos restrictivos, al número de muestras más restrictivo. Así se evitan pequeñas paradas de los motores, que impedirían el correcto funcionamiento del sistema

3.4 ESTABILIDAD DEL SISTEMA

No forma parte de los objetivos del presente trabajo el estudio detallado de la estabilidad del sistema. Las aproximaciones realizadas en el modelado del sistema, así como la utilización de valores estimados para la posición y velocidad de la pelota, unido a las saturaciones y no linealidades del sistema aumenta significativamente la complejidad del análisis. Tan sólo se ha tenido en cuenta que supuesto el sistema estable la ley de control diseñada obligue a que el error disminuya exponencialmente (ecuación (4)). La estabilidad asintótica se garantizará si no hay error en la medición y en la estimación, y se considera solo la cinemática del robot despreciando su dinámica (lo que es equivalente a decir que las velocidades articulares son ejecutadas instantáneamente por los actuadores).

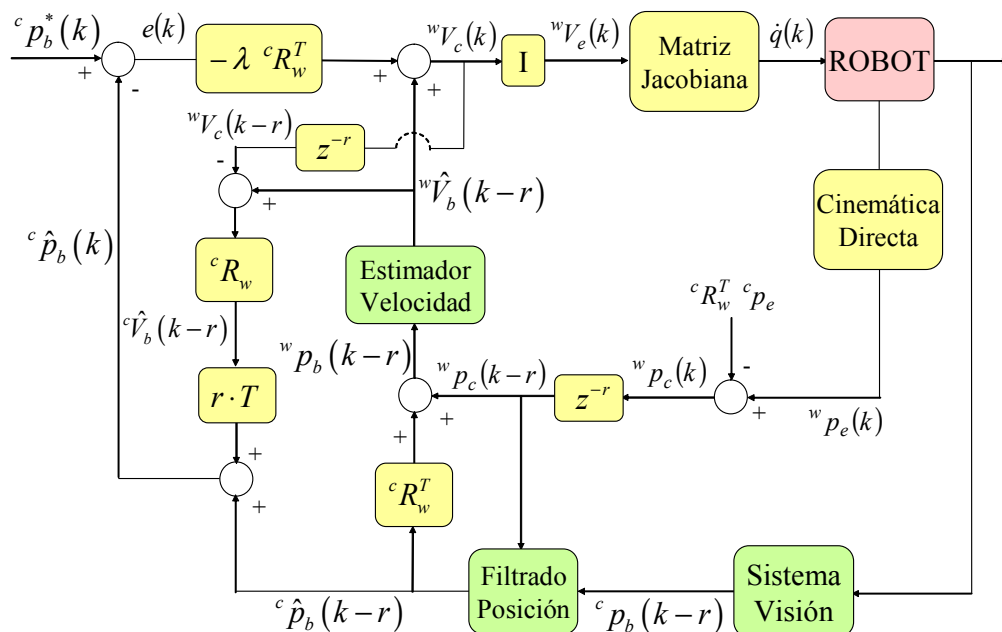


Figura 7. Arquitectura de control visual propuesta

4 RESULTADOS EXPERIMENTALES

En el presente apartado se describen las pruebas realizadas en el seguimiento de un objeto a alta velocidad (hasta 1 m/s) con el sistema Robotenis. El objetivo de control es que el extremo del robot permanezca a una distancia fija ($[0, 0, 600]^T$ mm) de la pelota. La pelota se sujeta a la estructura mediante una cuerda, y con un arrastre manual realiza distintas trayectorias. La Figura 8 representa la evolución tridimensional de la pelota en una prueba.

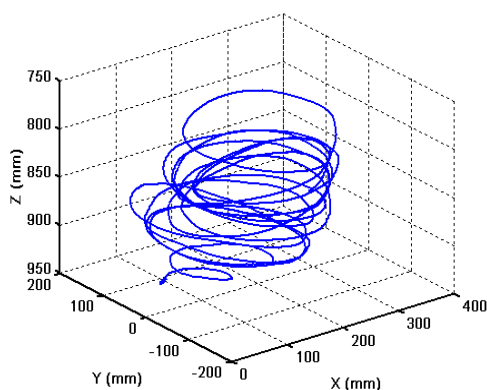


Figura 8. Evolución tridimensional de la pelota

4.1 INDICES DE SEGUIMIENTO

El caótico movimiento de la pelota en el extremo de la cuerda dificulta el estudio sistemático de las

pruebas, ante la dificultad para poder repetir las mismas efectuadas con distintos algoritmos de control. A tal fin se han definido dos índices de seguimiento basados en el error de seguimiento (diferencia entre la posición deseada de la pelota y la real) y la estimación de la velocidad de la pelota.

- *Relación de seguimiento*: Se define como la relación entre la media del módulo del error de seguimiento y la media del módulo de la velocidad estimada de la pelota. Permite aislar la influencia de la velocidad sobre el error de seguimiento. Se expresa en mm / mm/s

$$\text{Relaci3n de seguimiento} = \frac{\frac{1}{N} \sum_{k=1}^N \|e(k)\|}{\frac{1}{N} \sum_{k=1}^N \|\hat{w}_b(k)\|} \quad (17)$$

- *Media por franjas del módulo del error de seguimiento en función del módulo de la velocidad estimada de la pelota.* Los límites inferiores y superiores del módulo de la velocidad de la pelota son 0, 200, 400, 600, 800 y 1000 mm/s. Se expresa en milímetros

4.2 LEYES DE CONTROL ESTUDIADAS

Se han realizado pruebas con dos leyes de control:

- *Ley de control proporcional.* No se considera la componente predictiva de la ecuación (8).

$${}^w v_e(k) = -\lambda {}^c R_w^T [{}^c p_b^* - {}^c \hat{p}_b(k)]. \quad (18)$$

- *Ley de control predictiva.* Se considera la componente predictiva de la ecuación (8).

$${}^w v_e(k) = {}^w \hat{v}_b(k) - \lambda {}^c R_w^T [{}^c p_b^* - {}^c \hat{p}_b(k)]. \quad (19)$$

Las Tabla 1 y la Tabla 2 representan los resultados obtenidos para los anteriores índices cuando se aplican las dos leyes de control. Los datos numéricos se han obtenido haciendo las medias de 10 pruebas para cada algoritmo. Se observan claramente un mejor comportamiento de la estructura predictiva, que ocasiona una menor relación de seguimiento y un menor error por franjas.

Tabla 1. Relación de seguimiento para las leyes de control proporcional y predictiva

ALGORITMO	RELACIÓN DE SEGUIMIENTO
Proporcional	40.45
Predictivo	20.86

Tabla 2. Error por franjas para las leyes de control proporcional y predictiva

Velocidad	V<200	200-400	400-600	600-800	V>800
Algoritmo Proporcional	6.3	13.7	20.1	26.2	32.5
Algoritmo Predictivo	4.2	8.1	9.5	11.3	13.5

Las Figura 9 y Figura 10 muestran para un ejemplo concreto (para cada eje y en módulo) la evolución del error de seguimiento, la velocidad

estimada de la pelota, y la velocidad calcula para el extremo del robot, tanto para una ley de control proporcional como para la ley predictiva.

Para el control proporcional el máximo error de seguimiento es de 34.5 mm y la máxima velocidad detectada para la pelota es de 779.9 mm/s. Para el control predictivo el máximo error de seguimiento es de 18.1 mm la máxima velocidad detectada para la pelota es de 748.1 mm/s. Se observa como el empleo del término predictivo mejora significativamente el seguimiento.

5 CONCLUSIONES

El artículo presenta una novedosa estructura de control visual para un robot paralelo que tiene por objetivo interactuar con objetos que se mueven a alta velocidad con trayectorias desconocidas. Es el primer robot paralelo conocido en el que se realiza seguimiento visual en entornos dinámicos. La estrategia de control está basada en la predicción de la menor muestra en la que es posible cumplir la función objetivo. Se integran los retardos existentes en el sistema y las saturaciones producidas en la acción de control, lo cuál impide que el sistema intente alcanzar una velocidad superior a la permitida. Las pruebas realizadas muestran las altas prestaciones del sistema. Se realiza el seguimiento de una pelota de ping pong con un error inferior a 20 milímetros, y cerrándose el bucle visual en 8.33 ms. Como futuros trabajos se pretende alcanzar una velocidad de seguimiento cercana a los dos metros por segundo, que el sistema sea capaz de realizar nuevas tareas como impulsar a la pelota o agarrarla analizar la estabilidad del sistema.

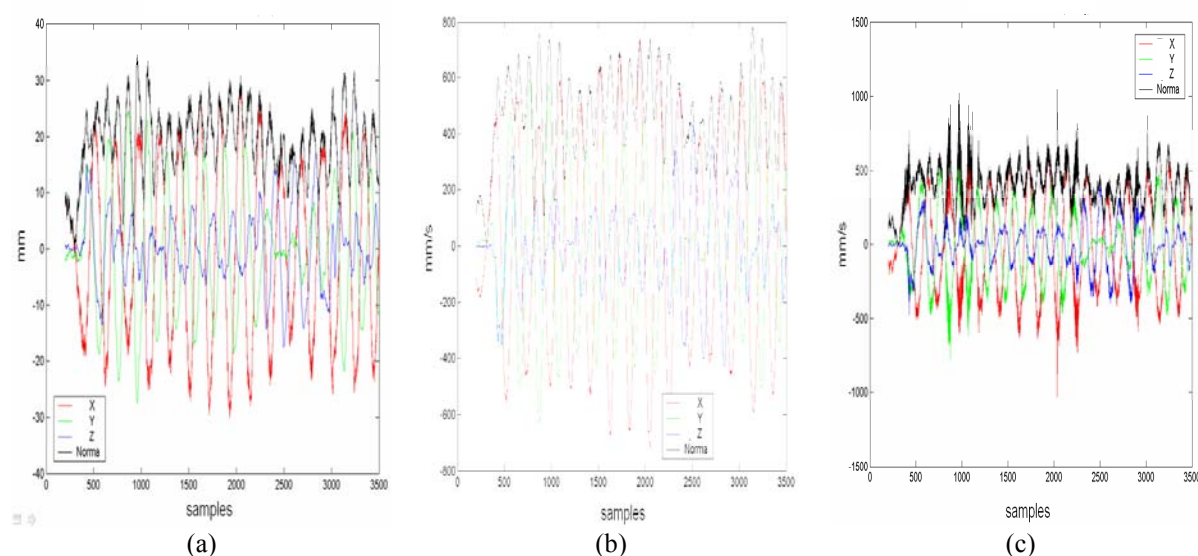


Figura 9. Ley de control proporcional. (a) Error de seguimiento. (b) Velocidad estimada de la pelota. (c) Velocidad calculada del extremo del robot

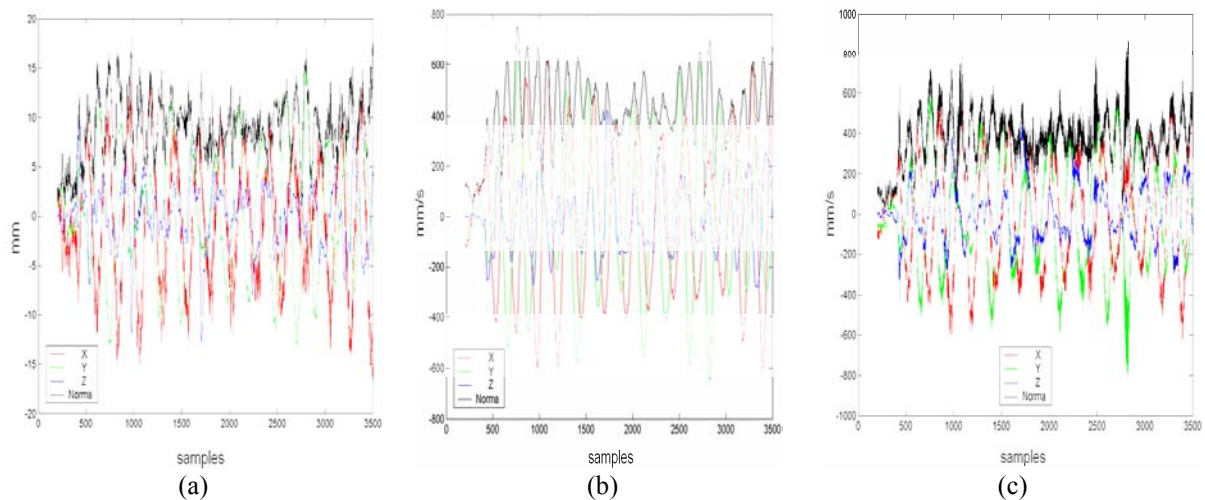


Figura 10. Ley de control predictiva. (a) Error de seguimiento. (b) Velocidad estimada de la pelota. (c) Velocidad calculada del extremo del robot

Referencias

- [1] Ángel, L. Control Visual de Robots Paralelos. Análisis, Desarrollo y Aplicación a la Plataforma RoboTenis. Tesis Doctoral de la Universidad Politécnica de Madrid. Diciembre 2005
- [2] Ángel, L., Sebastián, J.M., Saltarén, R., Aracil, R., Gutiérrez, R. (2005). RoboTenis: Design, Dynamic Modeling and Preliminary Control. IEEE/ASME AIM2005, 24-28 July 2005, Monterey, California USA.
- [3] Ángel, L., Sebastián, J.M., Saltarén, R., Aracil, Sanpedro, J. (2005). RoboTenis: Optimal Design of a Parallel Robot with High Performance. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2-6 August 2005. Alberta, Canadá.
- [4] Ángel, L., Sebastián, J.M., Saltarén, R., Aracil. RoboTenis System. Part II: Dynamics and Control. 44 IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05), Sevilla, 2005.
- [5] Gutiérrez, D. Estimación de la posición y de la velocidad de un objeto móvil. Aplicación al sistema RoboTenis. Proyecto Fin de Carrera de la E.T.S.I.I. de la Universidad Politécnica de Madrid
- [6] Gutiérrez, D., Sebastián, J.M., Ángel, L. Estimación de la posición y velocidad de un objeto móvil. Aplicación al sistema RoboTenis. XXVI Jornadas de Automática, 6-8 septiembre 2005, Alicante
- [7] Hutchinson, S.A., Hager, G.D., Corke, P.I. (1996): A tutorial on visual servo control. IEEE Trans. Robotics and Automation, 12-5 651-670.
- [8] Kaneko, M., Higashimori, M., Takenaka, R., Namiki, A., Ishikawa, M. The 100 G capturing robot - too fast to see. IEEE/ASME Transactions on Mechatronics. Volume 8, Issue 1, pp 37 – 44. March 2003
- [9] R. Clavel, DELTA: a fast robot with parallel geometry. 18th International Symposium on Industrial Robot, pp. 91-100, 1988, Sydney, Australia.
- [10] Kragic, D., Christensen, H.I. (2005). Advances in robot vision. Robotics and Autonomous Systems. 52 (1), 1-3
- [11] Senoo, T., Namiki, A., Ishikawa, M. High-speed batting using a multi-jointed manipulator. 2004 IEEE International Conference on Robotics and Automation, ICRA '04. Volume 2, pp 1191-1196, 26 Apr – 1 May, 2004.

APLICACIÓN WEB PARA VISUALIZACIÓN Y PROCESADO DE IMÁGENES BIOMÉDICAS

Manuel Torres, Roberto González, Enrique Torija, Gloria Bueno
 Universidad de Castilla-La Mancha, E.T.S.I. Industriales, Ciudad Real
 gloria.bueno@uclm.es

Resumen

La digitalización dentro del ámbito biomédico es ya un hecho, tanto en los dispositivos como en las aplicaciones, la transmisión y el almacenamiento de datos. Obtener el máximo rendimiento, haciendo uso de la tecnología disponible actualmente para compartir información, procesar y visualizar los datos diagnósticos depende de forma esencial de la creación de herramientas computacionales que integren modelos matemáticos y físicos que permitan obtener y describir a partir de dichos datos la información relevante para su interpretación en el diagnóstico clínico. En este trabajo presentamos una aplicación Web que integra herramientas para análisis de imágenes biomédicas, abarcando desde la visualización hasta el procesado. La aplicación ha sido diseñada y probada con imágenes de anatomía patológica (H&E, DO7, y TMA).

Palabras Clave: Visor de Patología, Procesado de Imágenes Biomédicas, Servicios Web, Anatomía Patológica.

1 INTRODUCCIÓN

La expansión de la World Wide Web (WWW) ha sido creciente en los últimos años, su capacidad de aplicación ha permitido invertir numerosos recursos para crear modelos y herramientas útiles a usuarios en diferentes áreas. La biomedicina no se queda al margen de estos avances, el uso de aplicaciones distribuidas basadas en servicios web, herramientas de colaboración, de compartición e interpretación de datos, como ayuda al diagnóstico es de gran utilidad en esta área. La utilización de estos servicios web para fines diagnósticos y educativos en biomedicina es posible gracias al desarrollo de sistemas de digitalización biomédicos [4, 5].

En este trabajo se ha hecho un estudio de las necesidades de especialistas médicos en Anatomía Patológica, relativas al diagnóstico mediante el análisis de imágenes en esta especialidad. De este estudio se concluye que los sistemas actuales de

imagen digital en Anatomía Patológica deben contemplar soluciones para la visualización y el procesado de la imagen en un entorno cooperativo, requisito para el desempeño eficaz del trabajo del patólogo [5]. A partir de este estudio se ha implementado una serie de algoritmos para la visualización y el procesado de imágenes patológicas todo ello integrado en una aplicación Web.

El objetivo, es por tanto, crear una aplicación Web accesible, eficiente y útil que permita visualizar imágenes patológicas de gran tamaño, que es uno de los grandes problemas que nos encontramos en la visualización de estas imágenes, pudiendo ser hasta 12 Gbytes de dimensión. La aplicación tiene que ser flexible, para soportar diferentes formatos de imagen (JPEG, JPG, TIF, GIF, BMP, PGM, PPM, PNG), y que a la vez posibilite la ejecución de distintos algoritmos de procesado de imágenes haciendo uso de la plataforma INBIOMED [3].

Dentro de estos objetivos, como se ha mencionado, nos centramos en el análisis de imagen. El análisis de imágenes biomédicas es ampliamente necesitado en esta área, que además puede requerir grandes necesidades de cálculo para obtener resultados satisfactorios. El uso de aplicaciones distribuidas basadas en servicios web es particularmente apropiado para estas necesidades tanto para la unificación de metodologías y sistemas existentes, como para la integración de nuevos métodos. Además, los sistemas PACS que normalmente se usan en centros hospitalarios y que integran imágenes en DICOM, permiten la selección y visualización de imágenes inter-departamentales, pero no tienen la capacidad de usar funciones de tratamiento digital de imágenes y aun menos compartición de recursos a nivel multi-céntrico. Sin embargo, mediante su interfaz de comunicación, sí podrían usar servicios web que conformen análisis de imagen, permitiendo trabajar con diferentes herramientas.

A continuación se explica la metodología utilizada tanto para el desarrollo del visor como para los servicios web de procesado de imágenes. Ilustrándolo con ejemplos de la aplicación web implementada.

Finalmente se exponen las conclusiones y futuras líneas de desarrollo de este trabajo.

2 MÉTODOS y RESULTADOS

El empleo de herramientas de análisis de imagen médica está cada vez más extendido dada la necesidad de tener métodos objetivos de evaluación e interpretación de las imágenes diagnósticas. En el proceso de evaluación de estas pruebas, el especialista está sujeto tanto a las limitaciones propias del ojo humano como a las limitaciones del sistema de digitalización y/o el sistema óptico. Diversos métodos han sido investigados y desarrollados para solventar estos problemas, pero en su mayoría son todavía costosos tanto computacional como económicamente. El procesamiento de imagen distribuido proporciona un entorno eficaz para la computación e integración de muchos datos y algoritmos diferentes. Las principales tecnologías que soportan procesamiento distribuido son: Java RMI (*Remote Method Invocation*), CORBA (*Common Object Request Broker Architecture*), DCOM (*Distributed Component Object Model*), RPC (*Remote Procedure Calls*), *Web Services Mobile Agents*, *GRID computing* [9].

En este trabajo se ha usado un método integrado basado en servicios web, para procesamiento de imagen distribuido dentro de la Red INBIOMED [3]. La principal ventaja de los servicios web radica en la sencillez con que se puede crear código que se ejecute en un servidor, pero que sea accesible desde un cliente remoto, además de la capacidad de integrar diferentes tipos de datos y modelos computacionales. La aplicación Web, aquí implementada, da soporte remoto a múltiples usuarios mediante un servidor (Apache Tomcat). Uso de servlets (Java) y de tecnología JSP. La metodología utilizada sigue:

- a) **Proceso Unificado:** Siguiendo las distintas fases del ciclo de vida del software.
- b) **Diseño Conceptual:** Utilizando UML, diagramas de casos de uso
- c) **Diseño Lógico:** Objetos de negocio y servicios. Diagrama de Clases UML dividido en capas.

La tecnología utilizada para el visor es:

- Applet (Java): HTTP sobre TCP en el puerto 80, por seguridad en empresas.
- HTTP Tunneling: Comunicación de applets con servidores remotos.
- HTTP Basic Authentication: Para autenticación en el cliente (login + password).
- Librería de Java para procesamiento avanzado de imagen.

para los servicios web de procesado:

- C, C++: Implementación de los algoritmos de procesado de imágenes (Eficiencia).

- .NET: Plataforma de integración con respecto a la aplicación web (Java).
- Plataforma INBIOMED: Integración a través del Lenguaje IQL.

El objetivo de la plataforma INBIOMED es:

- Integrar la información diagnóstica.
- Desarrollar un modelo de información común.
- Facilitar la utilización de herramientas de análisis.

A continuación se explica el desarrollo del visor de imágenes patológicas y los servicios de procesado de las mismas.

2.1 VISUALIZACIÓN DE LA IMAGEN

Las soluciones propuestas para la visualización de imágenes patológicas de forma eficiente han sido tratadas por las casas comerciales [2], y también han sido estudiadas por algunos grupos de investigación [6, 7, 8, 10]. El mecanismo habitual consiste en descomponer la imagen original en múltiples copias de distintas resoluciones, empezando por la resolución original hasta llegar a una resolución en tamaño *thumbnail*. A su vez cada capa es subdividida en pequeños fragmentos a modo de *grid*.

En estos sistemas para visualizar la zona de la imagen requerida, solamente se tendría que leer el fichero o ficheros, normalmente muy pequeños, con lo que la eficiencia es muy alta, y el tiempo de respuesta es prácticamente inmediato. La desventaja de este sistema, es la cantidad ingente de ficheros pequeños, que necesitamos, si la imagen es muy grande, como suele ser el caso de las imágenes de patología. Por ejemplo una imagen de 12 Gb, al fragmentarla con este sistema tiene unos 91.645 ficheros de dimensiones 256x256. Se pueden producir problemas al tener tantos ficheros, como son degradación del rendimiento del Sistema Operativo, el tiempo requerido para mover los ficheros, la portabilidad de los mismos, así como la posibilidad de que algún fichero se corrompa al salvarlos en los dispositivos ópticos de almacenamiento. Otra de las desventajas es que estos sistemas sólo proporcionan la visualización sin herramientas de edición (comentarios y anotaciones), herramientas que son particularmente necesarias en esta especialidad.

Se ha desarrollado una solución en este trabajo de investigación, que trata de solventar estos problemas. La metodología empleada para el desarrollo del visor corresponde al *Proceso Unificado*. El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a proyectos que varían en tamaño y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología de objetos en el desarrollo de software. Uno de los componentes clave es el UML (*Modelo Unificado de Lenguaje*).

Las fases que se han seguido para la implementación del visor han sido:

Análisis de los requisitos del usuario

La elaboración de Diagramas de Secuencia y de Casos de Uso a partir de una lista de requisitos, confeccionada en base a los requerimientos del usuario, ha supuesto el punto de partida para el desarrollo de la aplicación. En este sentido, se han mantenido una serie de reuniones con los usuarios para especificar qué funcionalidad debía cubrir el Visor.

Diseño de la aplicación

La aplicación está diseñada siguiendo un modelo de tres capas independientes, pero interconectadas:

- *Capa de Dominio*: representa el contenido lógico de la aplicación, pudiéndose decir que es la parte computacional de la misma.
- *Capa de Presentación*: está orientada a presentar la información al usuario.
- *Capa de Persistencia*: almacenará en un servidor, el contexto sobre el funcionamiento del visor.

Partiendo de este modelo de tres capas, se ha elaborado, en paralelo, el diseño de la base de datos, y las clases necesarias para la implementación de la capa de dominio.

Este diseño, como se ha apuntado anteriormente, emplea como base el Lenguaje de Modelado Unificado. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables.

2.1.1 Guía técnica de diseño

El visor se ha implementado como un Servicio Web para dar cabida, de forma remota, a los usuarios utilizando como “motor” el servidor de Apache Tomcat. El tratamiento de imágenes se ha considerado elemento primordial en el diseño. Es por ello que se ha seleccionado las librerías de Java. Este API permite un procesamiento de imágenes eficiente en *applets* y aplicaciones. Además posee un conjunto de operaciones que facilitan acciones como pueden ser el cambio entre el formato de imágenes o el almacenamiento de las mismas.

2.1.2 Almacenamiento de Imágenes

Implementado haciendo uso de tecnologías como *Servlets*, *J2EE* (Java 2 Enterprise Edition) y *jsp* (Java Server Pages). Se pretende conseguir, ante todo, eficiencia a la hora de subir las imágenes de

patología, teniendo en cuenta que normalmente serán de un elevado tamaño.

2.1.3 Visor de Imágenes de Patología

El Visor de Imágenes está diseñado como un applet de Java. Esta elección de diseño se debe a que normalmente el único puerto que los administradores suelen dejar abierto es el 80, por motivos de seguridad. Un problema común a la hora de desarrollar *applets* es la limitación de éstos a la hora de comunicarse con servidores Web. Para solucionar esta desavenencia el uso de *HTTP (Hypertext Transfer Protocol) Tunneling* permite a los applets la comunicación con sistemas remotos. De esta forma, la comunicación *applet* para el visor y los *Servlets* para el tratamiento de imágenes resultan eficientes.

También se ha considerado, como medida de diseño, el uso de autenticación en el cliente (HTTP Basic Authentication). Con esto se pretende que para poder acceder y utilizar el visor, el usuario sea reconocido, a través de un login y un password por el sistema.

Implementación de la Aplicación

Tras configurar el diseño del Visor, se ha procedido a la implementación y pruebas del sistema, siguiendo el paradigma de Programación Extrema (XP).

Implantación de la Aplicación

El paso definitivo es la implantación de la aplicación en la máquina que vaya actuar como servidor a través de Apache Tomcat. Tras la implantación de la aplicación por parte del administrador, el usuario puede utilizarla desde una máquina remota.

Las utilidades del visor, ilustradas en las Figuras 1 a Figura 7, son:

- Gestión de imágenes.
- Preprocesado de imagen: zoom, cambiar brillo y contraste.
- Edición: gestión de comentarios, mediciones de puntos de interés.



Figura 1: Inicio de la aplicación del visor



Figura 2: Gestión de imágenes del visor

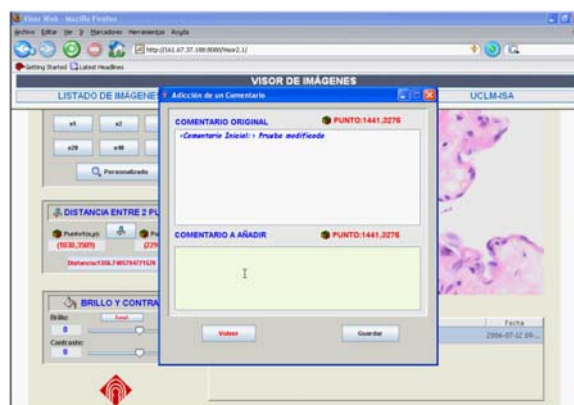


Figura 5: Inserción de comentarios

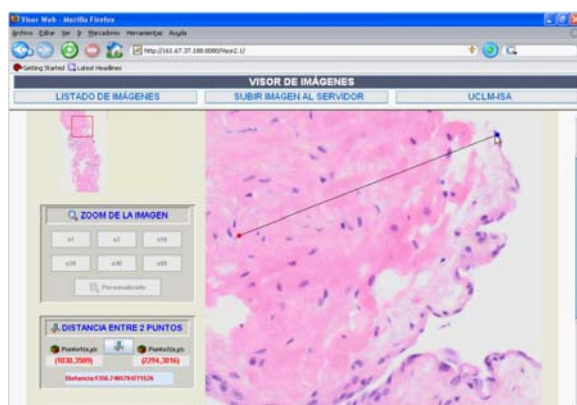


Figura 3: Visualización de la imagen a distinto zoom. Posibilidad de medir zonas de interés.



Figura 6: Gestión de comentarios

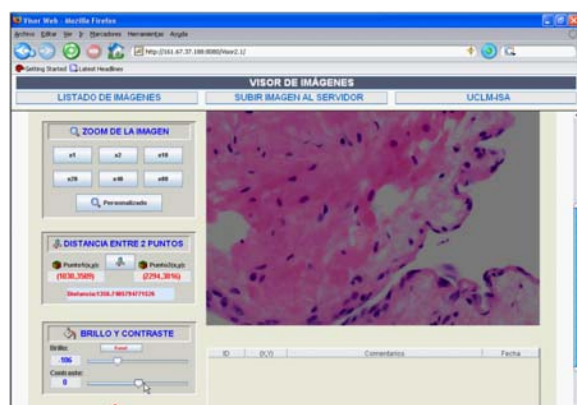


Figura 4: Preprocesado de la imagen

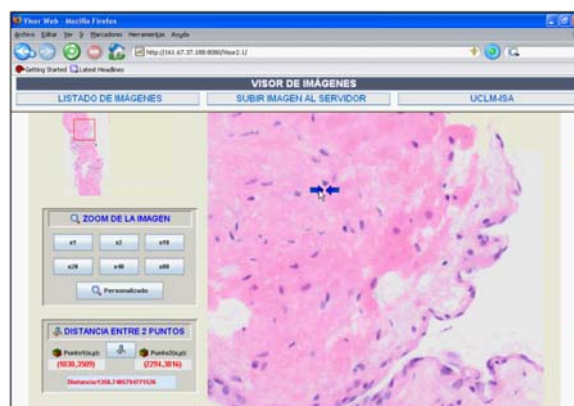


Figura 7: Recuperación de comentarios

Tanto la carga, como la visualización y preprocesado de una imagen se realiza de forma rápida en pocos segundos. El visor tiene la posibilidad de insertar comentarios, seleccionando la zona de interés donde se quiere hacer un comentario con el ratón. También es posible borrar y modificar dicho comentario (ver Figura 5 y Figura 6) y localizar el comentario en la posición y al aumento seleccionado (ver Figura 7).

El siguiente apartado explica los servicios web de procesado de imágenes implementados, que se integran en la plataforma, siguiendo la estructura que indica la Figura 8.

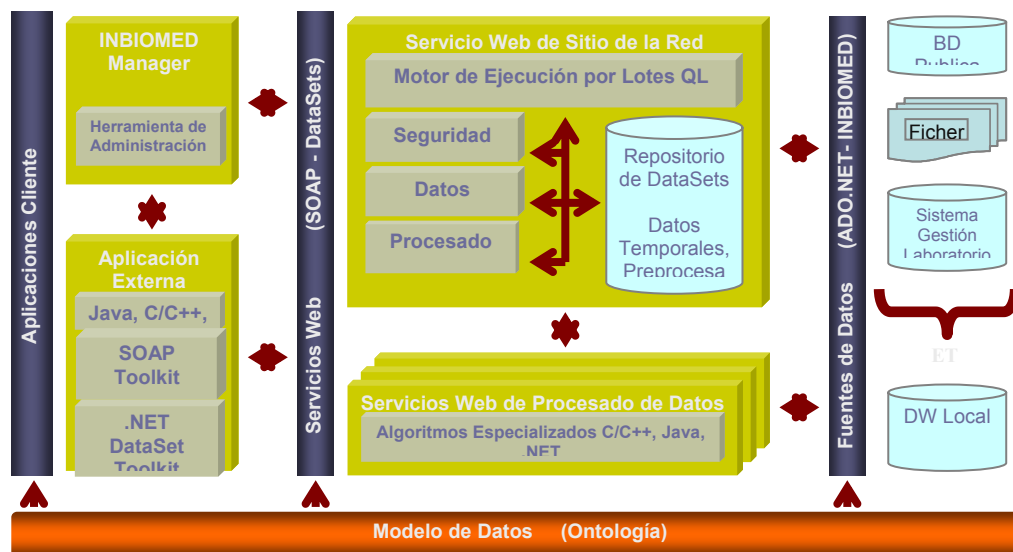


Figura 8: Estructura de integración de los servicios web en la plataforma INBIOMED

2.2 PROCESADO DE LA IMAGEN

Las herramientas de procesamiento desarrolladas se basan en:

- Selección del espacio de color.
- Detección de regiones de interés (ROIs) basado tanto en la homogeneidad de las ROI como en la heterogeneidad en los contornos.
- Descriptores de las ROIs para su posterior interpretación de las ROI detectadas.

Teniendo en cuenta el tipo de imágenes utilizados: molecular, funcional y genéticas. Concretamente el trabajo se ilustra con ejemplos de imágenes teñidas con hematoxilina-eosina (H&E) aplicado al estudio del carcinoma de próstata, con inmunohistoquímica para detección de p53 (DO7) en tumores de vejiga y con matrices de tejidos (TMA).

Los servicios web implementados hasta el momento cubren funcionalidades teniendo como restricción el tipo de imagen a tratar. Estas funciones se reflejan en la Tabla 1.

No entra dentro del marco de esta publicación detallar los métodos de procesamiento utilizados, el lector puede consultar [1], brevemente citaremos los aspectos de interés arriba mencionados.

Espacios de color

Después de un análisis exhaustivo de los diversos modelos de color: RGB, HSI, HSV y CIEL^a*b^{*} ó CIELuv, el modelo con el que mejor resultados hemos obtenido es el CIEL^a*b^{*}. Este es un espacio de percepción uniforme, basado en la teoría de

colores opuestos, donde la diferencia entre colores, tal y como lo percibe el ojo humano, se puede parametrizar mediante una distancia entre los colores, propiedad que no cumplen los espacios de color previos [11].

Los procesos de filtrado y realce de la imagen se han llevado a cabo de forma marginal, es decir, considerando independientemente cada canal. Por el contrario para la detección de ROIs se ha operado de forma vectorial, considerando el espacio como una terna o vector.

Detección de ROIs

La segmentación de ROIs se ha realizado teniendo en cuenta las discontinuidades producidas en los cambios de región, es decir en la detección de contornos y la homogeneidad que se tiene dentro de cada ROI. Esta homogeneidad ha permitido agrupar los píxeles de cada región por clusters utilizando lógica difusa [1] y técnicas basadas en derivadas parciales [2].

Interpretación de Regiones de Interés

Una vez detectadas las ROI, se debe caracterizar la prueba diagnóstica con una serie de características morfológicas y densitométricas, calculadas sobre las ROI. Así hemos calculado: la forma, el área, el número de núcleos, porcentaje de núcleos solapados, la tinción y grado. Actualmente se están analizando y añadiendo más descriptores para proporcionar una clasificación o indicio de benignidad o malignidad que ayude al diagnóstico de las imágenes.

Tabla 1: Servicios Web de Procesado de Imágenes.

Función	Tipo de Imagen
Area	Binarias (0 ó 255)
Umbralizar	En nivel de grises
Binarizar	En nivel de grises
Conteo Regiones	DO7, H&E y TMA
Contraste	DO7, H&E y TMA
Segmentación	DO7, H&E y TMA
Filtro Isotrópico	DO7, H&E y TMA
Gradiente	DO7, H&E y TMA
Infiltración	DO7, H&E
Negativo	DO7, H&E y TMA
Perímetro	Imágenes segmentadas
Forma	Imágenes segmentadas
Tinción TMA	TMA

Ejemplos de este procesamiento mediante los servicios web se ilustran en las siguientes Figuras 10-17. Previamente, se tiene que aceptar un certificado de seguridad, firmado digitalmente. Una vez aceptado aparecerá una pantalla como la de la Figura 9. A través de este cliente se ejecutarán los diferentes servicios web, mediante conexión a la plataforma INBIOMED por identificación del usuario.

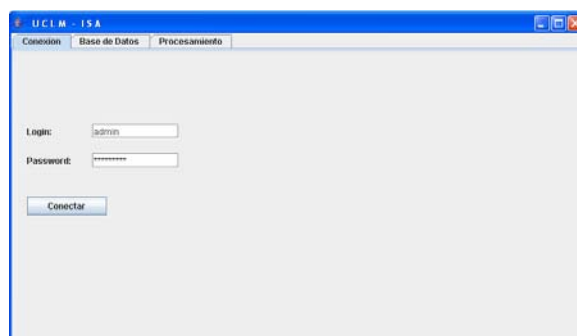


Figura 9: Conexión a los servicios web

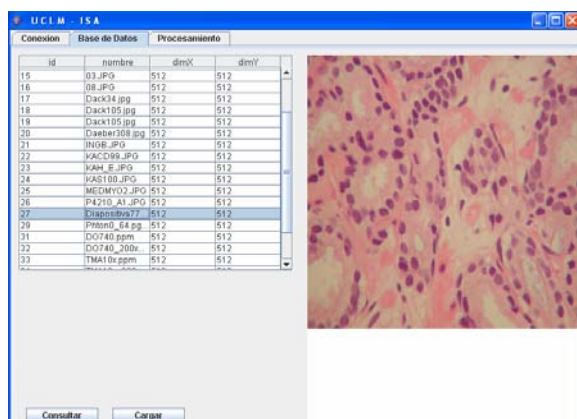


Figura 10: Consulta al servidor de la base de datos

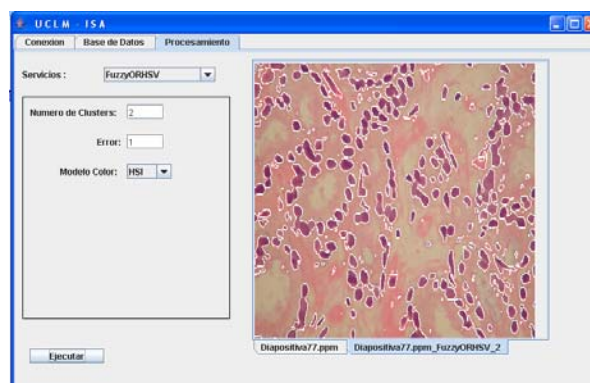


Figura 11: Procesado para la detección de células.

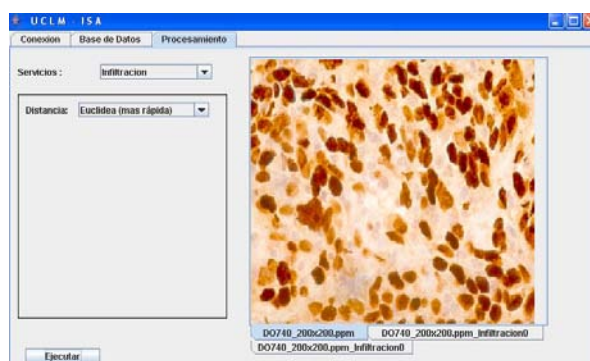


Figura 12: Procesado web en imagen de DO7

Se ha implementado una herramienta para detectar los porcentajes de las masas correspondientes a los núcleos que han quedado teñidas y no teñidas en imágenes de DO7. Para obtener estos porcentajes se realizan dos segmentaciones iterativas sobre los núcleos, mediante clusterización en el modelo de color CIE $L^*a^*b^*$ y aplicando distintas fórmulas de distancias (CIEDE2000 y Euclidea). Utilizando estos dos resultados se calculan los porcentajes de tinción. Todo este proceso es transparente al usuario y el único parámetro que debe especificar es el tipo de distancia que se utilizará en la segmentación. La diferencia entre la distancia Euclidea y CIEDE2000 está en el coste temporal y calidad del resultado; la distancia CIEDE2000 es más rápida pero desde el punto de vista temporal y computacional es mucho más compleja. Los resultados de esta herramienta son una imagen en la que se pueden ver a distintas intensidades las zonas que se han obtenido en la segmentación, es decir tinciones, (ver Figura 13) y un archivo en el que se muestran los porcentajes obtenidos de esta tinción (Figura 14).

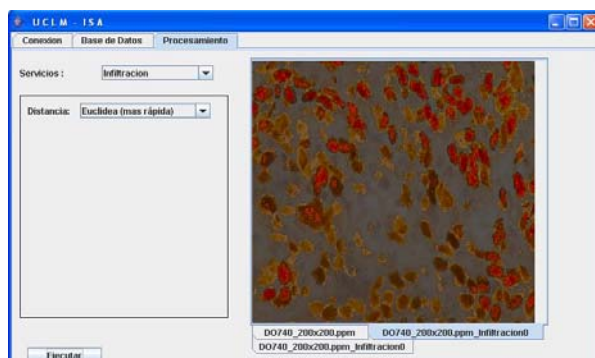


Figura 13: Infiltración en imagen de DO7

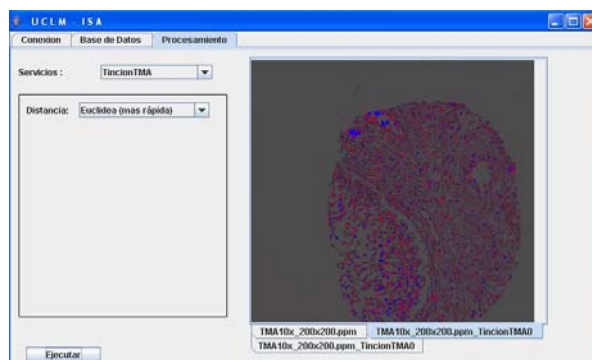


Figura 16: Tinción de núcleos en TMA

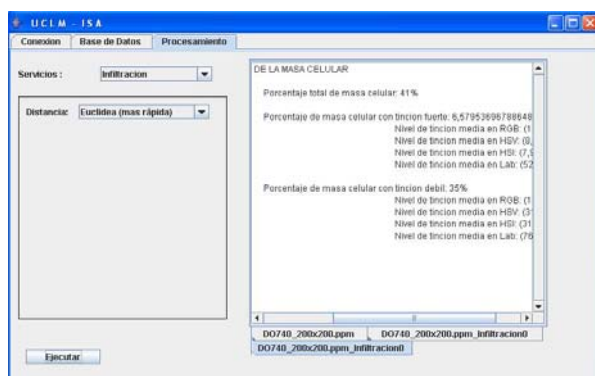


Figura 14: Porcentajes de tinción en imagen de DO7

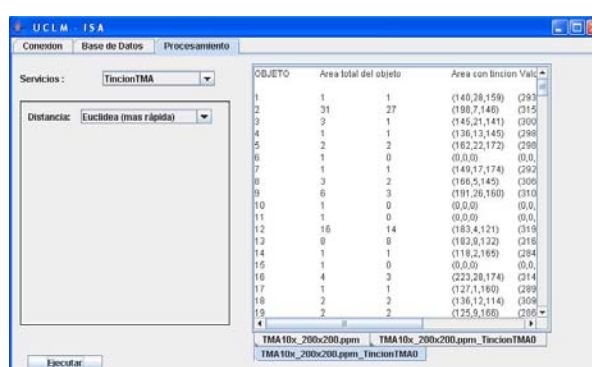


Figura 17: Porcentajes de tinción en imagen TMA

Otra de las herramientas implementadas es la detección de los niveles de tinción obtenidos en los distintos núcleos de la imagen de TMA (ver Figura 15). La segmentación se realiza en los modelos de color CIEL*a*b* y RGB analizando la tinción final de los núcleos en rojo y azul.

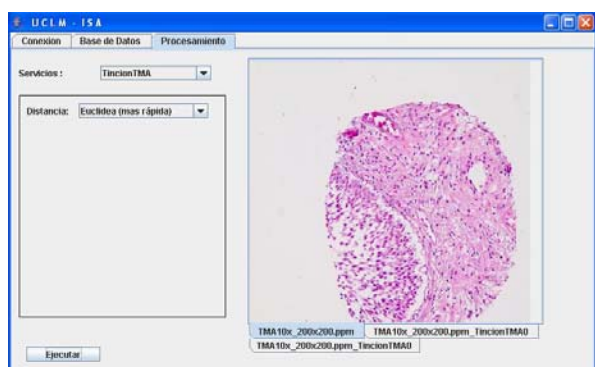


Figura 15: Procesado web en imagen de TMA

Los resultados de esta herramienta son una imagen en la que se pueden ver las tinciones R y B (ver Figura 16) y un archivo en el que se muestran los porcentajes de tinción en píxeles para cada uno de los núcleos (Figura 17).

3 CONCLUSIONES

En este artículo se ha mostrado la necesidad que hay dentro de la biomedicina y en particular en la especialidad de anatomía patológica, de tener herramientas de análisis de imagen, tanto para la visualización como para el procesado. El uso de estas herramientas de análisis y procesamiento de imágenes permitirán al patólogo tener una visión e información adicional que le facilitará la evaluación y diagnóstico de este tipo de imágenes.

Para la visualización, se ha presentado una herramienta desarrollada para visualizar imágenes de grandes dimensiones vía web. El visor ha sido probado por especialistas del Dpto. de Anatomía Patológica del Hospital General de Ciudad Real con buenos resultados visualizando imágenes de hasta 12Gb. Una de las ventajas es la posibilidad de introducir comentarios haciendo un entorno más cooperativo. En la actualidad se está ampliando con más opciones de gestión y procesado.

En el procesado, se han presentado diferentes técnicas, que muestran dar resultados satisfactorios para la segmentación de estructuras de interés en las imágenes analizadas y en entornos cooperativos. Las herramientas han sido evaluadas por usuarios de la red INBIOMED (IMIM), igualmente con resultados

satisfactorios. No obstante, la continuidad de este estudio con un mayor número de casos es necesaria para realizar un análisis estadístico más fiable y una posterior descripción de las características de las estructuras de interés en las imágenes diagnósticas, imprescindibles antes de concluir la validez de las mismas para la rutina médica.

Agradecimientos

Este trabajo ha sido realizado gracias a las ayudas de los proyectos del MEC (DPI2004-01346) y del FIS (red INBIOMED).

Referencias

- [1] Bueno G., González R., García M., (2005) "Procesamiento de imágenes histológicas aplicado al diagnóstico del carcinoma de próstata", *Actas del Simposio de Ingeniería de Sistemas y Automática en Bioingeniería*, SISAB2005, (CEA-IFAC), pp.119-124.
- [2] Bueno G., Martínez A., Adán A. (2004) "Fuzzy-Snake Segmentation of Anatomical Structures. *Lecture Notes in Computer Science*, 3212, pp. 33-42.
- [3] Estruch A., Heredia J.A., (2004) "Technological platform to aid the exchange of information and applications using web services", *Lecture Notes in Computer Science*, 3337, 458-468.
- [4] García M., Bueno G. González J. Carbajo M., (2005) "Preparaciones digitales en los servicios de Anatomía Patológica (I)", *Rev. Esp. de Anatomía Patológica*, 38: 69-77.
- [5] García M., Bueno G. González J. Carbajo M., (2005) "Preparaciones digitales en los servicios de Anatomía Patológica (II). Análisis de las soluciones existentes", *Rev. Esp. de Anatomía Patológica*, 38: 207-220.
- [6] Helin H., Lundin M., Lundin J., (2005) "Web-based virtual microscopy in teaching and standardizing Gleason grading", *Human Pathology*, 36:381-386.
- [7] Lundin M., Lundin J., Helin H. Isola J., (2004) "A digital atlas of breast histopathology: an application of web based virtual microscopy", *Journal of Clinical Pathology*, 57:1288-1291.
- [8] Marchesvsky A.M. Dulbandzhyan R., Seely K., Carey S., Duncan R., (2002) "Storage and Distribution of Pathology Digital Images Using Integrated Web-Based Viewing Systems", *Arch Pathology Lab Med.*, 126:533-539.
- [9] Pérez D., Crespo J., Anguita A., Dorado J., *et al.*, "Biomedical image processing integration through INBIOMED: A web services-based-based platform", *Lecture Notes in Computer Science*, 3745, 34-43.
- [10] Singh S., Kim D., y Mohler J. L., (2005) "Java Web Start based software for automated quantitative nuclear analysis of prostate cancer and benign prostate hyperplasia", *BioMedical Engineering OnLine*, 4:31.
- [11] Wyszecski G., Stiles W. S., (2005) *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd Edition. Ed. Wiley.

ANÁLISIS DE SISTEMAS EMPOTRADOS JAVA COMERCIALES CON REQUERIMIENTOS TEMPORALES

Juan A. Holgado, Alfredo Moreno, Jaime Viudez, Manuel I. Capel
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
{jholgado, amoreno, nieves, mcapel}@ugr.es

José Manuel Molina
REALTECH System Consulting, S.L.
jmanuelmontes@gmail.com

Resumen

Existen en el mercado una gama de sistemas empotrados comerciales en los que se utiliza Java como principal lenguaje de programación para el desarrollo de programas. El reclamo principal del uso del Java deriva del hecho de que, además de que se pueden desarrollar programas flexibles, robustos y seguros, presentan la propiedad de portabilidad que asegura su ejecución en diferentes plataformas hardware sin necesidad de recompilar los programas. Sin embargo, todas estas propiedades deseables para la programación tienen un precio que se paga en el rendimiento, en una falta en la predecibilidad de las tareas y en unas necesidades desmesuradas de memoria que no siempre se pueden admitir en aplicaciones que se tienen que ejecutar en sistemas empotrados con limitados recursos de procesamiento y memoria. En este trabajo se presentan las características principales de distintas propuestas comerciales de sistemas empotrados Java para el desarrollo de aplicaciones empotradas y de tiempo real. Cada una de estas propuestas solventa los problemas de Java de distinta manera añadiendo nuevos mecanismos y facilidades que no son siempre compatibles entre sí. Como resultado de este trabajo, se propone la utilización de una librería API de Java de propósito general que proporciona un modelo de programación común que puede ser utilizado en cualquiera de los sistemas empotrados analizados.

Palabras Clave: Sistemas empotrados, Java, J2ME.

1 INTRODUCCION

El desarrollo de dispositivos empotrados requiere de un alto nivel de integración entre sus componentes hardware y software para conseguir un producto de bajo costo por unidad, que permita su fabricación en serie [3]. Los recientes avances en hardware y la

reducción en su costo permiten reemplazar circuitos electrónicos complejos y procesadores costosos hechos a medida, por componentes más baratos y simples, que pueden ser reutilizados para otras aplicaciones. El desarrollador puede seleccionar aquellos componentes hardware más apropiados de acuerdo con sus necesidades de confiabilidad, costo, rendimiento, consumo de energía, tiempo de vida, etc. de entre una gama de microprocesadores de 8, 16 o 32 bits, DSPs (procesadores digitales de señales), o ASICs (circuitos integrados específicos para una aplicación). También tiene alternativas en cuanto a módulos de memoria de varios tipos, dispositivos de entrada y salida, contadores, etc.

Aunque los dispositivos empotrados proporcionan un ambiente programable sobre el cual es posible ejecutar el software diseñado, la mayoría de los sistemas empotrados aún se construyen utilizando una mezcla de lenguajes de programación C y ensamblador. De este modo, los desarrolladores pueden implementar programas más simples que logran un máximo rendimiento de acuerdo con las prestaciones del dispositivo, con un consumo mínimo de memoria y energía. Sin embargo, el uso de lenguajes procedimentales inseguros como C tiene un impacto negativo en el desarrollo de software, especialmente en el caso de aplicaciones complejas, que hace que el desarrollo del software se convierta en una tarea tediosa y propensa a errores en el que es muy difícil su mantenimiento y reutilización.

Java es un lenguaje de programación moderno orientado a objetos, que ha sido aplicado exitosamente para el desarrollo de aplicaciones para sistemas de escritorio y empresariales, aunque recientemente está ganando la atención de los desarrolladores de sistemas empotrados [10]. Java proporciona la infraestructura y los mecanismos necesarios para facilitar el desarrollo de aplicaciones que sean robustas, flexibles, seguras y reutilizables que, en teoría, pueden ser ejecutados en cualquier plataforma hardware gracias a la propiedad de

portabilidad. Por ello, Java puede ser una alternativa interesante para el diseño de sistemas empotrados, ya que puede simplificar el diseño, desarrollo, prueba y mantenimiento de aplicaciones complejas. Sin embargo, todas estas características deseables para el desarrollo de aplicaciones requieren sacrificar otras propiedades que pueden ser muy importantes para una aplicación concreta que se ejecuta en un sistema empotrado con recursos limitados en memoria y procesamiento como son el rendimiento, el determinismo temporal del sistema o el consumo de memoria. Se han adoptado varias posibles soluciones que solventan en parte algunas de estas dificultades, pero en ningún caso pueden considerarse como soluciones estándares.

En este trabajo se presentan las características de distintas propuestas comerciales de sistemas empotrados Java en los que se ha evaluado las posibilidades que ofrecen para el desarrollo de aplicaciones empotradas y de tiempo real. Cada una de estas propuestas proporciona una plataforma Java que siguiendo en parte algunas de las especificaciones estándares, imponen restricciones en la máquina virtual Java (JVM) donde se ejecutan las aplicaciones Java, y añaden nuevos mecanismos y facilidades para el desarrollo de aplicaciones empotradas incluyendo para ello nuevas librerías propias.

A diferencia de lo que suele suceder en otros trabajos [2][1], no se ha hecho un estudio del rendimiento o de las capacidades para el tiempo real de cada una de las propuestas, sino más bien se han analizado las librerías, los mecanismos, facilidades y abstracciones de alto nivel, así como las peculiaridades de la JVM para cada una de estas propuestas.

Cuando comparamos algunas de estas propuestas entre sí nos podemos encontrar diferentes mecanismos para efectuar las mismas funciones, por ejemplo en el uso de los temporizadores. Las divergencias entre las distintas librerías pueden originar incompatibilidades entre las distintas soluciones empotradas a las que el desarrollador de aplicaciones debe estar atento, dado que se pierde la propiedad de portabilidad de código. Es, por ello, que en este trabajo se propone el desarrollo de una nueva librería API genérica, HAJASE (Hardware Java para Sistemas Empotrados) para facilitar la portabilidad de programas para sistemas empotrados, es decir, diseñar programas que pueden ejecutarse para cualquiera de los dispositivos analizados. Esta librería consta de un conjunto de paquetes y clases que proporcionan un marco de trabajo uniforme, coherente y transparente para acceder a puertos de entrada y salida digitales o analógicos, interfaces de bus (I2C, CAN, 1-Wire, RS-232, RS-485, SPI), temporizadores y manejadores de interrupciones. El

desarrollo de esta librería tiene como origen un proyecto de domótica en el cual se buscaba la posibilidad de controlar los distintos dispositivos domóticos como sensores de presencia, controladores de persianas, climatizadores, lámparas, etc. a través de diferentes tipos de dispositivos empotrados Java [9].

2 JAVA PARA SISTEMAS EMPOTRADOS

Java se está convirtiendo en el lenguaje de programación preferido para muchos desarrolladores, ya que tiene muchas características interesantes que ayudan y facilitan el desarrollo, prueba y mantenimiento de aplicaciones, especialmente cuando éstas son complejas. Pese a todos los beneficios de Java como lenguaje de programación para el desarrollo de aplicaciones, existen algunas particularidades que pueden causar dificultades en el desarrollo de aplicaciones para sistemas empotrados. A continuación se analizan algunas de estas particularidades.

2.1 MODELOS DE EJECUCIÓN DE LA JVM

La máquina virtual de Java (JVM, Java Virtual Machine) es el núcleo del paradigma Java, y determina el rendimiento de la ejecución de los programas Java. Para ejecutar programas en Java primero deben ser compilados en un formato binario portable (en la forma de código intermedio neutro, llamado Bytecode) contenidos en ficheros .class, para que luego los bytecodes contenidos en dichos ficheros sean ejecutados en la JVM. La JVM no sólo traduce los bytecodes a instrucciones nativas, sino que además verifica todos los bytecodes antes de su ejecución. Existen cuatro modelos de ejecución de la JVM:

a) Interpretado: este es el modo clásico de ejecución, en el que la JVM interpreta bytecodes del fichero .class uno a la vez y ejecuta la operación o secuencia de operaciones nativas sobre la plataforma software del dispositivo (el entorno de ejecución del mismo). En este modo se satisface la característica de portabilidad de la filosofía Java, pero además se obtiene el peor rendimiento, como ejemplo en la JVM de las primeras ediciones del JDK.

b) Just-in-Time (JIT). El proceso de traducción de bytecodes a código nativo se realiza cada vez que un método se invoca o una clase se carga por primera vez. La secuencia correspondiente de código nativo se guarda en memoria RAM para una rápida ejecución la próxima vez que se llame al mismo bytecode. Este modo es más rápido que el anterior,

pero requiere una cantidad importante de memoria RAM para guardar tanto el código Java como su correspondiente código nativo. Existen variaciones del algoritmo JIT, como la compilación adaptativa dinámica (DAC) [4]. Las máquinas virtuales HotSpot de SUN y J9 de IBM son ejemplos de JVMs JIT.

c) Ahead-of-Time (AOT) o compilación cruzada: los bytecodes Java, o incluso el propio código fuente Java se traducen directamente a formato binario de la plataforma nativa del dispositivo, utilizando un compilador cruzado. Este método es similar al utilizado por aplicaciones en C/C++ con librerías compartidas. El fichero ejecutable que se obtiene puede incluir el fichero objeto más todas las librerías precompiladas en un solo fichero, o bien las librerías precompiladas pueden ser enlazadas en tiempo de ejecución. En cualquier caso, sólo puede ser ejecutado por el entorno de ejecución específico para el que fue construido. Con este modo se logra un rendimiento superior con menos uso de RAM, pero la portabilidad, la independencia entre plataformas y la flexibilidad de Java se sacrifican.

d) Procesador Java: este modo es el más rápido, ya que los bytecodes son interpretados y ejecutados directamente por el hardware del procesador, sin necesidad de traducción o de tan siquiera un sistema operativo subyacente. Existen dos formas de ejecutar bytecodes en hardware: con un procesador Java o con un chip acelerador [11][12]. En la primera, se reemplaza al procesador de uso general por un procesador hardware específico capaz de ejecutar bytecodes de JVM como su conjunto de instrucciones nativo, con lo que sólo puede ejecutar bytecodes Java. En cambio, en la segunda se incorpora un co-procesador junto al microprocesador de propósito general, el cual se encarga de traducir los bytecodes Java en secuencias de instrucciones para el procesador de propósito general. Con este tip, el mismo hardware pueda utilizarse para ejecutar código mixto. Existen varios procesadores Java con implementaciones diferentes de este enfoque, como el aJile JEM2, CJip, etc.

2.2 LIBRERÍA ESTANDAR

Sun ha agrupado su plataforma Java en tres ediciones, cada una orientada a un mercado particular: J2EE para aplicaciones empresariales, que requieren soluciones completas y escalables, como soluciones de comercio electrónico; J2SE para el mercado de ordenadores de sobremesa; y J2ME para dispositivos portátiles y empujados, incluyendo los dispositivos de gama baja, con limitaciones de recursos. Además, se creó una comunidad Java (Java Community Process) en la cual grupos especializados de compañías y organizaciones académicas pueden

desarrollar nuevas especificaciones de librerías (APIs) para campos de aplicación específicos y para las nuevas tecnologías que van surgiendo, como Bluetooth (JSR-82), tecnología inalámbrica (JSR-185), programación en tiempo real (JSR-1), Java TV (JSR-927), etc. En el mercado de los dispositivos empujados Java, se pueden encontrar productos que satisfacen alguna de las siguientes especificaciones:

- Java Card. La primera versión de esta especificación fue introducida en 1996, está orientada a pequeñas tarjetas inteligentes con procesadores de 8/16/32 bits y fuertes restricciones de memoria, en el rango de 1KB de RAM y 16 KB de ROM. [7]
- PersonalJava [13]. Introducida en 1997, es compatible con la versión de JDK 1.1.8, ahora discontinuada, está orientada a dispositivos empujados con 2,5 MB de ROM y un mínimo de 1MB de RAM. Actualmente esta especificación ha sido reemplazada por la de J2ME/CDC. Sin embargo, hay fabricantes que aún venden sistemas empujados con librerías que satisfacen esta especificación.
- EmbeddedJava. Introducida en 1998, es compatible con la versión discontinuada del JDK 1.1 orientada a dispositivos de gama baja con 512kB de ROM y 512 KB de RAM sin interfaces gráficas. Actualmente se ha reemplazado esta especificación por J2ME/CLDC.
- J2ME/CLDC [6]. Introducida en el 2000, su mercado natural son dispositivos con capacidad de conexión inalámbrica, como teléfonos móviles, buscapersonas, etc., y se ha convertido en el estándar de facto para la mayoría de los dispositivos empujados Java, incluyendo equipos de TV, etc. Requiere de al menos 192kB de memoria, y de un procesador de 16 o 32 bits.
- J2ME/CDC [6]. Introducida en el 2001, reemplaza a la especificación PersonalJava para los dispositivos empujados con al menos 2MB de memoria y procesadores de 32 bits. Se orienta principalmente a PDAs y a dispositivos empujados con más recursos.
- RTSJ [8]. Aparte del mundo de las aplicaciones empujadas, en el 2001 se introdujo una especificación para la programación de aplicaciones de tiempo real. Extiende el lenguaje Java, sin modificar la semántica del lenguaje para hilos que no tienen características de tiempo real, para que soporte tareas de tiempo real con planificadores que pueden ser modificados dependiendo del tipo de sistema de tiempo real. RTSJ fue diseñado para extender a la especificación J2SE (no a J2ME), por lo que uno de sus mayores problemas está en la gran cantidad de memoria que necesitan sus implementaciones, lo que dificulta su uso en

dispositivos empotrados, especialmente en aquellos de gama baja. Por ejemplo, la implementación de referencia de Timesys requiere 2,6 MB de memoria para la JVM además de 2,5 MB para las librerías.

El uso de la implementación de una librería API que sea conforme a una de las especificaciones descritas anteriormente sobre un sistema empotrado establece la semántica de Java y las posibilidades que ofrece en cuanto a la programación. Ahora bien, esto no es suficiente para la programación de un dispositivo empotrado específico, dado que existen otros aspectos importantes a considerar en la programación de dispositivos empotrados que se describen en la próxima sección.

2.3 OTRAS CARACTERISTICAS

Aunque las especificaciones Java mencionadas anteriormente definen y especifican como deben implementarse los programas Java en un entorno empotrado, la plataforma Java tiene algunos inconvenientes que tienen que tenerse en cuenta cuando se trabaja con sistemas empotrados.

El modelo de seguridad de Java no permite un acceso directo al hardware del dispositivo para el que se programa, debido a que Java carece de punteros, sólo referencias. Por tanto, no es posible controlar directamente el contenido de los registros hardware, ni de las posiciones de memoria, lo cual hace imposible la implementación de controladores para controlar dispositivos de entrada/salida. Sólo es posible acceder al hardware utilizando métodos nativos Java a través de la extensión JNI (Java Native Interface) que nos permite invocar llamadas a procedimientos externos escritos en otros lenguajes de programación como C o ensamblador. Sin embargo la eficiencia de este mecanismo es muy dependiente del entorno de ejecución y tiene un rendimiento muy pobre. Por este motivo la mayoría de los fabricantes de dispositivos empotrados aportan sus propios interfaces a métodos nativos para mejorar el rendimiento global de las aplicaciones.

La plataforma Java estándar proporciona un soporte muy débil para aplicaciones de tiempo real, debido a que mecanismos dinámicos de Java como la carga dinámica de clases, el enlazamiento dinámico o el recolector de basura, son no deterministas por naturaleza. Además la semántica de Java para el manejo de múltiples hilos de ejecución y de tiempo real es débil y no está suficientemente especificada. En la mayoría de las ocasiones esto se resuelve incluyendo en el entorno de ejecución un sistema operativo de tiempo real sobre el que se ejecuta una máquina virtual de Java que utiliza el modelo de

hilos nativos, es decir, hace corresponder los hilos de Java con los hilos del sistema operativo. Sin embargo, esto no es suficiente para garantizar que tenga un comportamiento en tiempo real.

Los fabricantes de empotrados superan algunas de las debilidades de Java desarrollando nuevas librerías que complementan los mecanismos y medios que se le ofrecen a los desarrolladores de sistemas empotrados. Por tanto, por una parte, los fabricantes anuncian la compatibilidad de sus dispositivos empotrados con especificaciones Java estándar (como PersonalJava) y por otro lado incluyen librerías propietarias para completar el entorno de desarrollo de software para los dispositivos.

3 DISPOSITIVOS EMPOTRADOS JAVA

En esta sección se ofrece una descripción de algunos dispositivos empotrados Java comerciales. Los dispositivos seleccionados cumplen con las siguientes restricciones. Primero, pueden incluir procesadores de 8, 16 o 32 bits, pero deben ser soluciones comerciales con un costo por unidad conocido a través de sus sitios Web o distribuidores. En segundo lugar, el lenguaje de programación principal utilizado para dichos dispositivos debe ser Java. En tercer lugar, los dispositivos no deben pertenecer al dominio de los dispositivos móviles con conectividad inalámbrica, como PDAs, buscapersonas o teléfonos móviles.

3.1 JAVELIN STAMP

El Javelin Stamp consiste en una placa de tamaño reducido, diseñada por Parallax Inc., para hacer simple el desarrollo y despliegue de pequeños sistemas prototipo. Incluye un microcontrolador RISC (Ubicom SX48AC), convertidores de señal analógica a digital (A/D) y digital a analógica (D/A) y pines de entrada y salida (E/S) genéricos, que pueden ser utilizados para la implementación de temporizadores, comunicación serial o para los convertidores. La familia de microcontroladores SX además introduce el concepto de Periférico Virtual (VP, Virtual Peripheral), un método para utilizar una parte del tiempo de procesamiento del microcontrolador para realizar funciones de periféricos en software, funciones que normalmente se llevarían a cabo por chips externos; módulos como el de comunicación serial y los temporizadores se implementan como periféricos virtuales, manejados con librerías propietarias proporcionadas por el fabricante. El kit de desarrollo de la placa Javelin también incluye elementos como un regulador de voltaje, una placa de prueba (proto-board) para

implementar pequeños circuitos, etc., todo para facilitar las pruebas y permitir el desarrollo rápido de pequeños prototipos.

La placa Javelin Stamp utiliza un intérprete Java para traducir los bytecodes Java a instrucciones nativas del microcontrolador SX. La placa soporta un subconjunto de la especificación Java 1.2; sin embargo, la implementación de las clases y métodos es muy diferente de la implementación estándar. Entre las diferencias más importantes, no tiene soporte multihilo, no incluye recolección de basura, incluye solo un subconjunto de tipos primitivos de datos y de clases estándar, los arrays de caracteres solo pueden contener caracteres ASCII y no Unicode, no hay clases del tipo interfaz, y los arrays están limitados a una dimensión.

El paquete stamp.core es la librería que contiene los objetos y métodos para manejar los recursos de hardware en forma de objetos y periféricos virtuales. Una clase muy interesante en este paquete es la stamp.core.CPU, que contiene llamadas específicas para ayudar al programador a manejar los recursos del procesador, como los pines de E/S o el puerto serial. Para escribir, compilar, descargar y ejecutar programas en la Javelin se utiliza el entorno de desarrollo (IDE) de la Javelin Stamp, proporcionado por el fabricante. Este IDE incluye además un depurador de programas que se ejecutan dentro del mismo dispositivo.

Las limitaciones de esta placa en cuanto a recursos de hardware y librerías Java disponibles hacen difícil el desarrollo de aplicaciones complejas. Además, el código que resulte de este desarrollo no será portable a otras soluciones empuetradas Java. No obstante, esta placa es una buena alternativa para el prototipado rápido de pequeñas aplicaciones de prueba, y es factible el desarrollo de pequeñas aplicaciones de tiempo real utilizando el modelo de ejecutivo cíclico.

3.2 AJILE

El aJile JEMcore es un procesador Java de ejecución directa, que puede ser instalado como un procesador principal nativo Java para un dispositivo empuetrado. Existen 2 versiones de este procesador, el aJ-80 y el aJ-100; El procesador aJ-100 tiene soporte para 2 JVM, con lo que se pueden ejecutar hasta 2 aplicaciones independientes entre ellas, con sus propios intervalos de tiempo de ejecución y protección de sus espacios de memoria. Dentro de estos intervalos y espacios de memoria, el usuario puede especificar para cada JVM las políticas de uso de memoria y ejecución multihilo independientemente.

Pueden encontrarse en el mercado varias placas basadas en estos procesadores aJile. Entre ellos tenemos la JPro de Quest Innovations, JStamp y JStik de Systronics Inc. Todos ellos incluyen una UART para la comunicación serial, soporte para buses SPI e I2C, y puertos de E/S genéricos. Opcionalmente, pueden incluir controladores para los protocolos Ethernet, bus CAN, IrDA o 1-Wire.

El aJile JEMCore ejecuta directamente los bytecodes de máquina virtual Java y bytecodes adicionales para operaciones específicas del empuetrado. Con ello se incrementa significativamente el rendimiento en la ejecución de programa, con lo que los bytecodes se ejecutan como instrucciones nativas, el rendimiento del procesador JEMCore es similar al de un procesador RISC ejecutando código C compilado [5], y es posible desarrollar aplicaciones de tiempo real con él, aunque no es conforme con RTSJ.

El aJile JEMCore incluye un entorno de ejecución Java conforme con la especificación J2ME-CLDC. Se incluyen todas las clases de CLDC más clases adicionales dentro del paquete com.ajile, para permitir el manejo de las características particulares del dispositivo, como por ejemplo la clase PeriodicTimer en com.ajile.components, que permite la configuración de un temporizador con métodos abstractos.

Los programas escritos para los dispositivos basados en procesadores Ajile deben escribirse en un ordenador para luego ser descargados a la tarjeta, como en el caso de la Javelin. aJile proporciona dos herramientas software para realizar la descarga a dispositivos aJile: JEMBuilder y Charade. El entorno de ejecución del aJile no soporta la carga dinámica y enlace dinámico de clases, por lo que los programas descargados al aJile deben tener enlazadas todas sus clases antes de su ejecución; ésa es la tarea principal de JEMBuilder. El programa Charade se utiliza para descargar el fichero binario producido por el JEMBuilder al dispositivo y para efectuar labores de depuración.

El proceso de desarrollo y despliegue de aplicaciones en el caso de tarjetas aJile no es tan simple y directo como en el caso de la tarjeta Javelin, ya que los programadores tienen que realizar un paso extra de compilación, lo que implica la instalación y manejo de herramientas software específicas para ello. Además, no es posible probar y depurar las características de tiempo real de las aplicaciones desarrolladas utilizando Charade. Adicionalmente, se han encontrado problemas con el hardware de la tarjeta JPro, específicamente en el controlador de Ethernet y puertos serie, los cuales causan bloqueos y fallos inesperados en la ejecución de los programas,

que se comportan entonces de manera no determinista.

3.3 TINI

TINI (Tiny InterNet Interface) es una especificación de plataformas empujadas desarrollada por Dallas Semiconductor (ahora Maxim Integrated Products). El hardware de TINI consta al menos de los siguientes integrados: un microcontrolador, memoria ROM flash, y RAM estática. La familia de microcontroladores utilizados en las implementaciones de la especificación TINI es la DS80Cxx. Tiene soporte de comunicación serial, bus 1-Wire y bus CAN. La especificación TINI no obliga a los fabricantes a un único diseño de hardware para las placas, con lo que tienen libertad y flexibilidad para integrar dispositivos TINI en muchos sistemas; sin embargo, Dallas Semiconductor ha desarrollado una implementación de referencia, la placa TINI 390.

La ROM de las placas TINI contienen el código nativo con el que se implementa el sistema operativo subyacente (TINI OS), la máquina virtual Java, y las clases de Java. El entorno de ejecución Java de la especificación TINI combina clases de varios paquetes definidos en el JDK 1.1.8 de Sun; sin embargo, no soporta toda la especificación, por lo que no es conforme con ella. Faltan características importantes como la carga dinámica de clases y la finalización de objetos.

Dallas Semiconductor, como otros fabricantes de dispositivos empujados, ha desarrollado un número de paquetes y clases específicas para TINI (en el paquete `com.dalsemi`, clases como `I2CPort` en `com.dalsemi.system`), con métodos de alto nivel para manejar las comunicaciones con el bus I2C.

Para crear, compilar y desplegar una aplicación software para un sistema empujado TINI son necesarias varias herramientas software, disponibles en Internet. El kit de desarrollo de software de TINI (TINI SDK), de Dallas Semiconductor, incluye el sistema operativo TINI OS y la máquina virtual. Para compilar un programa Java para TINI, puede utilizarse cualquier compilador y cualquier entorno de desarrollo Java, siempre y cuando se utilicen las librerías del API de TINI para compilar los programas. Sin embargo, los empujados TINI no pueden ejecutar ficheros Java (.class) directamente, como en el caso del aJile, por lo que hay que llevarlos a un formato binario más compacto (.tini), utilizando un convertidor incluido en el TINI SDK, llamado TINIconvertor. Con ello se pierde buena parte de la portabilidad del código final.

3.4 SNAP

SNAP (Simple Network Application Platform) es una placa de Imsys Tech basada en el procesador Java Cjip. Se acopla a una placa portadora que provee hardware adicional, como conectores de Ethernet y de protocolo serie. Además, es totalmente compatible a nivel hardware, y hasta cierto punto a nivel software, con la especificación TINI. La placa incluye memoria ROM flash y RAM; tres controladores UART para comunicación serial, controlador Ethernet, soporte para I2C, SPI, CAN y 1-wire, además de pines de pines de E/S genéricos. El procesador Cjip soporta múltiples conjuntos de instrucciones nativas, lo que permite que coexistan lenguajes como Java, C, C++ y ensamblador en el desarrollo de software. Incluye un sistema operativo de tiempo real, llamado Moose, el cual maneja los temporizadores, interrupciones, hilos y mensajes de control.

El microprocesador de la SNAP, como en el caso del aJile JEMCore, ejecuta directamente bytecodes Java sin necesidad de un traductor o intérprete. El Cjip soporta y es conforme con J2ME-CLDC. Incluye la implementación del perfil MIDP. Imsys ha desarrollado sub-paquetes de clases para dar soporte específico al hardware de la placa SNAP, dentro del paquete `se.imsys`, con clases como `se.imsys.system.RTC`, que permite manejar el reloj de tiempo real con métodos de alto nivel. Además, incluye y soporta las librerías del paquete `com.dalsemi` de TINI.

La programación y descarga de aplicaciones en el caso del SNAP es un proceso más fácil y flexible que en el caso del aJile o del TINI, en parte porque el sistema operativo del procesador Cjip, Moose, permite directamente el acceso al empujado a través de una sesión telnet, y descargar las aplicaciones directamente mediante FTP. Además, no hay necesidad de realizar un paso o una compilación adicional para llevar los programas de Java a un formato binario especial, como en el caso de aJile o TINI, ya que Cjip puede ejecutar bytecodes Java generados por cualquier compilador o entorno de desarrollo.

3.5 EJC

EJC (Embedded Java Controller) es una plataforma empujada Java de uso general diseñada por Snijder Micro Systems, la cual está orientada hacia aplicaciones Java con conexión a Internet, al igual que TINI. El hardware de la placa consiste de un controlador de la familia EC200, basado en el procesador EP7312 de Cirrus Logic. La placa además

incluye controladores para Ethernet, interfaces serie, buses I2C y 1-Wire, y puertos de E/S de propósito general. Incluye un sistema operativo de tiempo real, llamado Intent,, con un planificador con desalojo. Además tiene una implementación de JVM. Intent y su JVM siguen un modelo de ejecución JIT, en el cual los bytecodes de Java deben ser primero traducidos a código nativo (a través de la JVM) antes de su ejecución, como en el caso de TINI y Javelin Stamp. La traducción se realiza de manera estática, cuando se compilan las aplicaciones, o dinámicamente, cuando se cargan clases nuevas.

La versión de Java de EJC soporta y es conforme con la especificación PersonalJava, versión 1.2. Adicionalmente, Snijder incluye el paquete com.snijder para manejar características de bajo nivel con clases abstractas, como se.imsys.system.PortIO para manejar las entradas y salidas. El kit de desarrollo de aplicaciones de EJC (ADK) de Snijder es una plataforma de desarrollo que permite descargar y ejecutar aplicaciones Java en el dispositivo empujado. Como en el caso de TINI OS, el sistema operativo Intent permite a los programadores iniciar una sesión con el empujado con telnet, descargarlas aplicaciones con FTP, y además crear macros (scripts) para la ejecución de programas. Para el desarrollo de aplicaciones Java para el EJC, como en el caso del SNAP, pueden utilizarse cualquiera de los muchos editores de texto, entornos de desarrollo y compiladores disponibles.

4 LIBRERÍA HAJASE

4.1 DESCRIPCION GENERAL

La librería HAJASE es una nueva librería Java de alto nivel que proporciona un marco de trabajo uniforme, coherente y transparente para manejar los mecanismos y facilidades de bajo nivel que contienen los distintos empujados analizados en este trabajo. La librería desarrollada no implementa ninguna solución nueva para el manejo a bajo nivel de empujados desde Java, sino que aprovecha los distintos mecanismos utilizados por los fabricantes en los diferentes empujados estudiados para definir una API de alto nivel que facilita el trabajo con cualquiera de ellos con un interfaz común.

La librería no requiere la utilización de ningún tipo de configuración o perfil de J2ME. Funciona sobre cualquier implementación de Java que incluya las clases básicas de java.lang y java.util. Las funcionalidades que cubre la nueva librería son:

- Acceso al estado a los puertos de entrada/salida digital o analógica.
- Soporte de comunicaciones para buses digitales como I2C (1-Wire y SPI próximamente).

- Servicios para el control y manejo del tiempo como temporizadores, timeouts, relojes del sistema, etc.
- Soporte para el manejo de interrupciones (no terminado todavía).
- Soporte para comunicaciones industriales RS-232, RS-485 y CAN.

4.2 DISEÑO DE LA LIBRERÍA

Para abstraer el hardware y las características específicas de cada empujado se sigue una filosofía basada en la orientación a objetos. La librería proporciona una interfaz al programador consistente en varias clases con métodos abstractos (VCPU, VTIMER, BUSI2C, VCOM). La clase VCPU virtualiza el procesador con todas sus entradas y salidas digitales y analógicas. La clase VTimer es una clase para el manejo de temporizadores que se encarga de la gestión de activaciones periódicas o aperiódicas de procesos en nuestras aplicaciones que pueden asociarse además a objetos que pueden representar a las diferentes salidas digitales y analógicas manejadas por la clase VCPU. También permite obtener el tiempo en milisegundos, así como producir retardos (delays) en la clase VCPU. La clase BUSI2C generaliza un Bus I2C proporcionando un conjunto de operaciones básicas (lectura y escritura) sobre los dispositivos que están conectados al bus I2C. Por último la clase VCOM se encarga de ocultar los detalles de la comunicación serie RS-232 del empujado con el exterior.

Para el diseño de la librería hemos utilizado el patrón de diseño **Factoría** que nos permite inicializar las clases VCPU, Vtimer, BUSI2C, VCOM con la versión especializada dependiente del empujado subyacente. Esto nos permite por una parte desacoplar totalmente la interfaz de la librería respecto de las versiones especializadas de la misma a la vez que nos permite una abstracción del hardware para los programadores que utilicen la librería, ya que manejarían directamente la clase abstracta con la implementación concreta.

Como ejemplo, podemos ver en la figura 1, el diagrama de clases de uno de los paquetes de la librería HAJASE que virtualiza el procesador del sistema empujado. El mecanismo de funcionamiento del patrón Factoria hace que al llamar al método estático getCPU de la clase Factoria CPU pasando como parámetro el nombre del empujado, el método se encarga de instanciar un objeto de la clase especializada concreta al empujado indicado.

Para reducir el tamaño que ocupa el programa Java que utilice Hajase, se han desarrollado scripts que se encargan de realizar la compilación del programa

Java para cada sistema empotrado añadiendo de Hajase sólo las clases concretas específicas de dicho sistema empotrado.

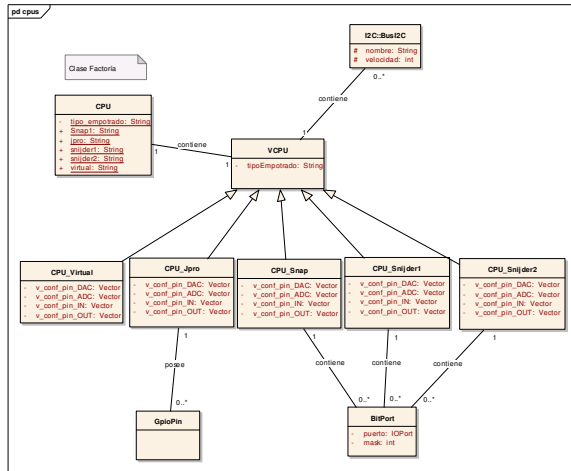


Figura 1. Diagrama de clases del paquete CPU de la librería Hajase.

5 CONCLUSIONES Y TRABAJO FUTURO

Java constituye una plataforma muy potente para el desarrollo de aplicaciones empotradas con requerimientos estrictos de rendimiento y prestaciones. Sin embargo, existe una gran cantidad de especificaciones, mecanismos y modelos de programación particulares a cada fabricante de dispositivos, lo cual dificulta la adopción de Java por parte de los desarrolladores de sistemas empotrados. En este trabajo se han expuesto las características principales de una muestra significativa de las soluciones Java empotradas más populares, y se propone una librería de alto nivel, basada en los mecanismos abstractos proporcionados por los distintos fabricantes, que permite manejar de forma homogénea y coherente los recursos de bajo nivel de todos los dispositivos analizados. En el futuro se prevé expandir la librería desarrollada para cubrir mecanismos y tecnologías aún no soportadas, y realizar un análisis del rendimiento de aplicaciones que utilizan esta librería para realizar las optimizaciones oportunas.

Referencias

[1] Chen, G. (2001) "PennBench: a benchmark suite for embedded Java", *Proceedings. WWC-5. 2002 IEEE International Workshop on Workload Characterization*, pp. 71-80.

[2] Corsaro, A., (2002) "Evaluating real-time Java features and performance for real-time embedded systems", *IEEE Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings.*, pp. 90-100.

[3] Debardeleben, J.A., (1997) "Incorporating cost modeling in embedded-system design", *Design & Test of Computers, IEEE*, pp. 24-35.

[4] Debbabi, M., (2005) "Accelerating embedded Java for mobile devices", *Communications Magazine, IEEE*, pp 80-85.

[5] Hardin, D.S., (2001) "Real-Time Objects on the Bare Metal: An Efficient Hardware Realization of the Java™ Virtual Machine", *Proceedings. 4th International Symposium on Object-Oriented Real-Time Distributed Computing*, pp. 53-59.

[6] Java 2 Micro Edition, Sun Developer Network, Sun Microsystems, <http://java.sun.com/javame>

[7] Java Card Documentation, Sun Microsystems, <http://java.sun.com/products/javacard/>

[8] Java SE Real-Time, Sun Microsystems, <http://java.sun.com/javase/technologies/realtime.jsp>

[9] Montes, J.M., Viudez, J. (2005) "Sistema de Control Domótico", Proyecto fin de Carrera. Universidad de Granada.

[10] Nilsson, A., (2001) "Deterministic Java in Tiny Embedded Systems", *Object-Oriented Real-Time Distributed Computing, 2001. ISORC-2001. Proceedings. Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, pp. 60-68.

[11] Schoeberl, M., (2004) "Restrictions of Java for Embedded Real-Time Systems", *Object-Oriented Real-Time Distributed Computing, 2004. Proceedings. Seventh IEEE International Symposium on Object-oriented Real-time Distributed Computing*, pp. 93- 100.

[12] Schoeberl, M., (2005) "JOP: A Java Optimized Processor for Embedded Real-Time Systems", Memoria de Tesis Doctoral, Vienna University of Technology.

[13] PersonalJava Application Environment, Sun Microsystems, <http://java.sun.com/products/personaljava/>

DISEÑO DE UN SISTEMA DE CONTROL DOMOTICO BASADO EN JAVA

Juan A. Holgado, Jaime Viudez, M. Capel
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
{jholgado, nieves, mcapel}@ugr.es

José M. Molina
REALTECH System Consulting, S.L.
jmanuelmontes@gmail.com

Resumen

En el presente trabajo se describe el diseño e implementación Java de una solución domótica abierta, modular y flexible que facilita la integración de los diferentes dispositivos domóticos presentes en una vivienda, el acceso al sistema desde diferentes protocolos de comunicación tanto cableado como inalámbrico, así como la posibilidad de utilizar desde ordenadores o portátiles a dispositivos móviles como teléfonos móviles o Pdas. Para conseguir esa integración se ha adoptado una arquitectura cliente-servidor multicapa que facilita tanto la sustitución de componentes como la extensión a nuevos componentes. Por otra parte se han desacoplado las tareas de acceso, monitorización y control de los dispositivos domóticos respecto de las peticiones concurrentes de los clientes que demandan servicios domóticos, intercalando un subsistema que actúa de pasarela entre los clientes y los dispositivos domóticos, lo que nos puede garantizar la fiabilidad del sistema a la vez que se obtiene un comportamiento más determinístico en las tareas de control. Además permite mejorar la escalabilidad del sistema cuando sea necesario controlar un número cada vez más mayor de dispositivos domóticos. Para poder comprobar el funcionamiento del sistema, hemos construido además un modelo físico a escala que contiene gran parte de los dispositivos domóticos de una vivienda, muchos de los cuales se han tenido que diseñar y construir específicamente para el sistema como el climatizador o las persianas. En el trabajo se describen también algunos de los detalles de construcción de la maqueta.

Palabras Clave: Java, Domótica, Modelo a escala de una vivienda, Automatización de sistemas, Integración de dispositivos móviles.

1 INTRODUCCIÓN

El avance vertiginoso de las tecnologías informáticas, electrónica, automática y de las telecomunicaciones

está produciendo un sinnúmero cada vez mayor de nuevos productos de consumo “automáticos” que se están incorporando poco a poco en todas las facetas de nuestra vida cotidiana con el compromiso de proporcionar servicios hasta hace poco tiempo inimaginables en un plazo de tiempo relativamente corto para así satisfacer y mejorar nuestra calidad de vida. Dentro de este vasto mundo tecnológico que nos invade, la domótica integra al conjunto de las tecnologías y sistemas automáticos que se utilizan dentro del entorno del hogar y, que proporcionan servicios de gestión energética, seguridad, bienestar y comunicación además de permitir una mejor conservación y cuidado de la vivienda.

Aunque todavía la domótica no ha tenido el grado de implantación y aceptación esperado, debido principalmente a la complejidad, falta de seguridad y altos costes derivados de su instalación y puesta en funcionamiento, se han desarrollado numerosas propuestas, muchas de las cuales son comerciales, basadas en la utilización de estándares domóticos en cuanto a tecnologías específicas del sector como Havi, Osgi, etc, como en el uso de redes domóticas como X-10, Lonworks, EIB, etc [3].

En este trabajo se presenta una solución domótica implementada completamente en Java que facilita al usuario el acceso, monitorización y control de forma local y remota de todos los dispositivos de la vivienda. Para ello proporciona interfaces de usuario flexibles e intuitivos para que el usuario final pueda manejar de forma sencilla el sistema, y además permite el acceso uniforme y simultáneo al sistema desde cualquier ordenador o portátil que se conecte a Internet, desde dispositivos móviles como el teléfono móvil o un PDA utilizando medios de transmisión inalámbricos y desde una centralita con pantalla táctil que se puede ubicar en la propia casa. Es un diseño propio abierto, modular y flexible que no sigue en principio ningún estándar ni protocolo de comunicaciones domótico, aunque gracias a su arquitectura es factible la integración de nuevos protocolos de comunicación, así como la

incorporación de nuevos dispositivos domóticos adaptando y/o extendiendo algunos de los componentes software ya desarrollados, lo que nos permite reducir además el tiempo de desarrollo de los nuevos componentes.

Para probar el funcionamiento del sistema desarrollado hemos construido una maqueta física a escala que contiene una gran parte de los dispositivos domóticos que puede haber en una casa real. Cada uno de estos dispositivos domóticos compuestos por sensores o actuadores y la electrónica necesaria para que sea controlables desde una unidad de control, se han diseñado y construido específicamente para este sistema como componentes hardware que pueden ser replicados para las diferentes habitaciones de la maqueta o reutilizados para la construcción de dispositivos domóticos más complejos. En las secciones siguientes se describen algunos de los aspectos más relevantes que se han tenido en cuenta tanto en su construcción como para su control desde la unidad de control.

2 DESCRIPCIÓN DEL SISTEMA

2.1 CARACTERÍSTICAS PRINCIPALES

El sistema de control domótico (SCD) realizado en este trabajo es un sistema abierto, flexible y modular, en el que cada componente desarrollado tanto a nivel hardware como a nivel software se ha efectuado buscando la extensibilidad del sistema, es decir, posibilitando la adición de nuevos dispositivos domóticos para lo cual se pueden adaptar parte de los componentes ya desarrollados, reduciendo así el esfuerzo en el desarrollo. Por tanto, es técnicamente posible desarrollar actualizaciones en el software o en el hardware sin producir un impacto que afecte negativamente al funcionamiento del sistema, e incluso puede realizarse dicho proceso de forma dinámica teniendo algunos de los módulos del sistema en ejecución.

En este tipo de sistemas suele ser habitual operar con numerosos sensores y actuadores de diferente naturaleza física con los cuales se sondea, monitoriza e interactúa con el entorno del hogar. Por tanto, el sistema debe permitir la escalabilidad del mismo manteniendo tanto la fiabilidad como el desempeño del sistema.

El sistema trabaja a diferentes escalas con diferentes requerimientos temporales según su cercanía al hardware. Cuando un cliente necesita activar un regulador como puede ser el modulo detector de presencia, requiere que tanto la captura del estado del entorno como las acciones de control a considerar se efectúen a una frecuencia determinada, no

admitiendo retrasos en la entrega de la acción correctora o en el envío de alarmas.

Una de las características importantes del sistema al que se ha prestado especial atención, es la interoperabilidad del sistema en cuanto a las redes de comunicaciones con el que los clientes pueden acceder al sistema de control domótico, así como la posibilidad de trabajar de forma común con diferentes tipos de microcontroladores utilizando Java como lenguaje de referencia para el desarrollo del software de control. Un cliente, por tanto, puede acceder al SCD utilizando desde ordenadores a dispositivos móviles como PDAs y teléfonos móviles a través de redes cableadas por Ethernet utilizando TCP/IP o redes inalámbricas a través Wifi o Bluetooth.

Desde el punto de vista de la facilidad de uso el sistema ofrece a los usuarios un entorno GUI sencillo e intuitivo adaptado al ordenador o dispositivo móvil utilizado. Por tanto, un usuario que utilice un móvil con una pantalla de reducidas dimensiones podrá utilizar un interfaz gráfico adaptado al mismo que permite manejar el sistema con el uso de dos o tres teclas, mientras que el usuario que trabaja desde un ordenador dispondrá de un interfaz mucho más rico sobre el que podrá interaccionar de la forma habitual a través de menús, formularios, etc.

Funcionalmente el sistema domótico va a llevar a cabo el control de los siguientes módulos:

a) *Control de Iluminación.* El sistema permite llevar a cabo un control de iluminación independiente en cada una de las habitaciones y en el exterior.

b) *Gestión de Seguridad.* El SCD desarrollado incorpora un sistema de seguridad que se aplica en varios niveles:

- Control de presencia, que consiste en varios sensores de presencia por infrarrojos distribuidos en cada una de las habitaciones.
- Sistema de detección de gas y humo consistente en un sensor de humo que monitoriza la presencia de gases en la cocina para evitar posibles escapes de gas.
- Simulación de presencia como elemento de seguridad pasivo en el que el sistema interactúa sobre los dispositivos domóticos diseminados por la vivienda, y los activa o desactiva siguiendo un patrón que puede ser almacenado previamente.

c) *Control de persianas.* El sistema también permite llevar a cabo el control de varias persianas, permitiendo al usuario subir y bajarlas completamente o bien moverlas paso a paso en cualquiera de los dos sentidos.

d) *Control de Temperatura (Climatizador)*. El sistema permite llevar a cabo el control de la temperatura mediante un climatizador en una de las habitaciones.

e) *Gestión de avisos y alarmas*. El Sistema permite a los usuarios activar o desactivar cada sensor y así mismo consultar su estado en cualquier instante. Cuando cualquiera de estos sensores detecte una situación de riesgo se disparan en el SCD diversos mecanismos de aviso a los usuarios:

- Aviso sonoro y luminoso en las terminales de los clientes conectados al sistema.
- Aviso sonoro mediante una sirena en el interior de la vivienda.
- Aviso a todos los inquilinos de la vivienda mediante el envío de un mensaje de correo electrónico
- Aviso a todos los inquilinos mediante el envío de un SMS.

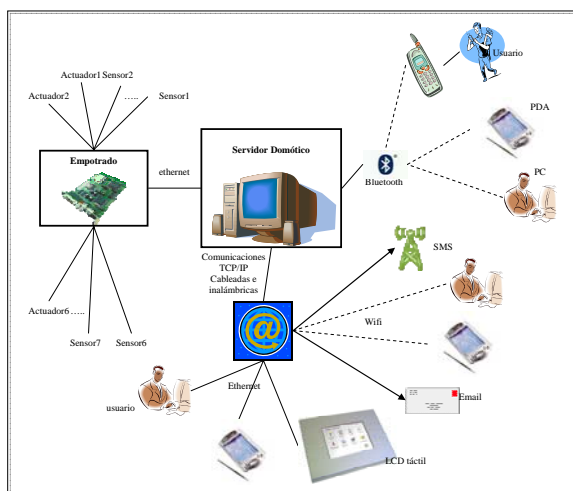


Figura 1. Componentes del sistema SCD.

2.2 COMPONENTES DEL SISTEMA

El Sistema de Control Domótico (SCD) está formado por tres subsistemas bien diferenciados e interconectados entre sí con distintos grados de responsabilidad y confiabilidad en función de las tareas a realizar (figura 1):

a) *Subsistema de control o empotrado* encargado del acceso, monitorización y control de los dispositivos domóticos (sensores, actuadores, ...) a más bajo nivel, así como de ejecutar las tareas de automatización más simples como regulación de iluminación o de climatización, notificación de avisos y alarmas que disparan eventos asíncronos sobre el SCD, etc. El subsistema de control se puede ejecutar en varias unidades de control. Cada unidad de control es un sistema empotrado con un microcontrolador de baja prestaciones que incluye el software apropiado para llevar a cabo tanto labores

de monitorización del entorno como de activación de reguladores simples sobre la parte del conjunto de dispositivos domóticos que controla directamente. El software de control que se ejecuta en las unidades de control utiliza una librería genérica Java (HAJASE, Hardware+Java para Sistemas Empotrados) que hemos desarrollado para dispositivos con limitados recursos de memoria y procesamiento, que uniformiza el uso de estos sistemas y permite un acceso transparente a ellos [2]. De este modo puede sustituirse una unidad de control por otra en tiempo de ejecución sin necesidad de desarrollar nuevo software, siempre y cuando haya una implementación específica de la librería para dicha unidad de control. Aunque hemos probado varias unidades, en el prototipo final del SCD utilizamos una única unidad de control para controlar a todos los dispositivos domóticos.

b) *Servidor domótico* se ejecuta en un ordenador independiente de la unidad de control y se encarga de gestionar los usuarios que pueden acceder al sistema SCD desde diferentes medios de transmisión como, por ejemplo, conexiones de red cableadas Ethernet a través de TCP/IP o conexiones inalámbricas a través de Bluetooth o Wifi, y se comunica con el subsistema de control para tratar las peticiones que les hagan los diferentes clientes que se conectan simultáneamente al sistema. Por tanto, el subsistema actúa como intermediario entre los clientes y el subsistema de control del SCD. Hay varias razones para separar la responsabilidad del SCD en el Subsistema Servidor domótico del Subsistema de control. En primer lugar, la unidad de control dada sus limitaciones no debe añadir a sus funciones propias de control y de acceso a los dispositivos domóticos, otras tareas de gestión de alto nivel como la gestión de usuario mencionada anteriormente. La diversificación de tareas puede saturar la unidad de control y ocasionar una pérdida en las prestaciones del mismo. En segundo lugar, la unidad de control tiene que satisfacer unas restricciones temporales más estrictas para el control, que no son necesarios en tareas de alto nivel. Por tanto, el sistema debe presentar diferentes niveles de predecibilidad según el tipo de tarea a ejecutar; en las tareas de control no debe producirse ninguna pérdida en los límites temporales impuestos a las acciones de control y consulta del estado de los sensores, mientras que para la gestión, acceso o envío de peticiones a usuario es factible una pérdida en el tiempo requerido para la entrega de una respuesta sin que esto represente un problema crítico al sistema. En resumen, el servidor domótico actuaría como la pasarela residencial del sistema domótico SCD, que se encarga de encauzar todas las peticiones externas de los usuarios a la red interna del sistema domótico sin provocar la saturación del Subsistema de Control para que pueda trabajar con características de tiempo real.

c) *Subsistema de clientes*. Este subsistema incluye todas las posibles formas en las que una aplicación cliente puede demandar servicios del servidor domótico. Los diferentes tipos de cliente dependen del tipo de sistema de procesamiento utilizado desde dispositivos móviles o PDAs a ordenadores PC y del tipo de medio de transmisión utilizado por el cliente pasando desde conexiones cableadas como inalámbricas (figura 2). Los tipos de clientes que admite el sistema, en resumen, son:

- Un cliente que utiliza un ordenador PC o portátil que se conecta al Servidor Domótico por Ethernet o Wifi.
- Un cliente que utiliza un ordenador PC o portátil que se conecta al Servidor Domótico mediante Bluetooth.
- Un cliente que utiliza un PDA para conectarse al Servidor Domótico por Wifi.
- Un cliente que utiliza un PDA o un teléfono móvil para conectarse al Servidor Domótico por Bluetooth.
- Un cliente que utiliza la pantalla táctil de la centralita para acceder al Servidor Domótico por Ethernet.

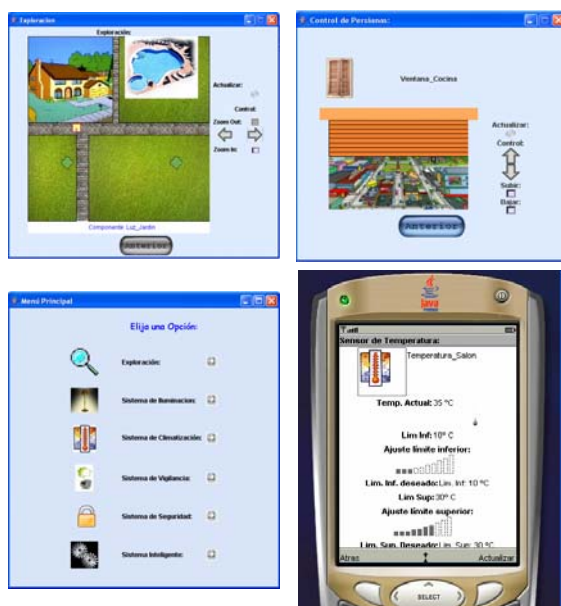


Figura 2. Captura de pantallas del subsistema cliente para diferentes dispositivos.

Por tanto, el sistema se divide en tres grandes subsistemas que se ejecutan de modo independiente en nodos distintos, de modo que entre cada par de subsistemas se establece una comunicación cableada o inalámbrica siguiendo una arquitectura cliente-servidor. Los subsistemas clientes (móvil, PDA, PC, ...) realizan peticiones de servicios domóticos al subsistema Servidor Domótico utilizando distintos tipos de protocolos de comunicaciones cableados o inalámbricos. Por otra parte, el subsistema Servidor Domótico actúa como cliente en la comunicación que se establece con el Subsistema de Control, que actúa como servidor del subsistema Servidor Domótico.

3 DESCRIPCIÓN HARDWARE

DEL

Para cumplir con los propósitos que nos hemos planteado de reutilización de todos los componentes del sistema, es necesario aplicarlos tanto a los elementos hardware y software del sistema. A continuación se describen algunos aspectos del hardware.

3.1 CONSTRUCCIÓN DE LA MAQUETA Y DISPOSITIVOS DOMÓTICOS

La construcción de la maqueta se ha realizado teniendo en cuenta los elementos del sistema de control domótico que queríamos automatizar en la casa, con la intención de obtener un modelo físico final ampliable y/o adaptable a nuevos usos futuros de modo que pueda ser utilizado para fines docentes. Para ello se ha elaborado una maqueta fácilmente desmontable para facilitar las labores de mantenimiento y reparación, ocultando en la mayor medida de lo posible el cableado que pasa por las habitaciones y proporcionando la suficiente movilidad para que pueda ser desplazada sin excesivos problemas. En la figura 3 se pueden ver algunas imágenes del prototipo final del modelo físico.



Figura 3. Imágenes del modelo físico

No se ha utilizado ningún dispositivo domótico real para la realización de este proyecto, ya que, en general, tienen grandes dimensiones y no pueden ubicarse en la maqueta. Por tanto, hemos tenido que diseñar y construir muchos dispositivos domóticos a escala. Para que sean controlables por el SCD es necesario conseguir niveles de tensión e intensidad adecuados por lo que se han incorporado circuitos amplificadores y estabilizadores. Los dispositivos domóticos (sensores y actuadores) manejados desde la unidad de control en el prototipo final del SCD se resumen en la tabla 1.

Tabla 1. Sensores y Actuadores utilizados en el prototipo final del SCD

Sensores	Actuadores
1 Sensor de Humo: a través de un ADC conectado por I2C	6 Bombillas: a través de un DAC por I2C cada una.
1 Sensor de Temperatura analógico: A través de un ADC conectado por un bus I2C	1 Climatizador (célula de peltier): a través de un driver de corriente de alta potencia con 2 patillas digitales.
8 Sensores Pulsadores para el control de persianas: a través de una entrada digital.	4 Motores DC para el control de persianas: a través de drivers de corriente (2 patillas digitales cada uno)
6 Sensores de Luminosidad: a través de un ADC conectado por I2C cada uno.	1 Sirena. Se conecta al sistema a través de un relé usando una patilla digital.
6 Sensores de Presencia: a través de una entrada digital cada uno.	

Todos los circuitos electrónicos se han diseñado y construido buscando la modularización y su posible reutilización para distintos componentes del sistema. Este tipo de sistemas es muy sensible a las interferencias que pueden aparecer en las señales lógicas, ya que hay un gran número de dispositivos en el sistema que trabajan con señales analógicas que se comparan con una tensión de referencia. Para disminuir estos efectos indeseables se han aislado en la mayor medida de lo posible los dispositivos perturbadores del resto del sistema mediante optoacopladores.

3.2 ELECCIÓN DE LA UNIDAD DE CONTROL O EMPOTRADO

La unidad de control del subsistema de control del SCD se encarga de realizar un control de bajo nivel de todos los dispositivos electrónicos (sensores, actuadores, reguladores, amplificadores, etc) necesarios para el buen funcionamiento del SCD. La unidad de control utilizada va a ser un microcontrolador o un sistema empotrado conectado a los dispositivos domóticos del SCD, que se encarga de generar las señales eléctricas precisas para leer el estado de los sensores, lo que nos permite monitorizar el entorno físico del hogar, y por otra parte genera las señales de control adecuadas para accionar los actuadores que interaccionan sobre el entorno físico del hogar.

Para conectar los dispositivos domóticos al sistema empotrado podemos o bien utilizar los puertos GPIO (entradas/salidas de propósito general) con el que podemos realizar un control directo de los mismos o bien utilizar buses digitales como pueden ser I2C, SPI ó 1-Wire para ampliar la conectividad del empotrado aumentando el número de dispositivos a

los que podemos tener acceso. Los sistemas empotrados cuentan en general con un número escaso o inexistente de entradas/salidas analógicas y digitales, sobre todo analógicas. Para ampliar el número de puertos de E/S hemos optado por realizar circuitos que utilicen conversores ADC y DAC para entradas y salidas analógicas, y expansores digitales para E/S digital, que se conectan a la unidad de control a través de un bus digital. En el prototipo final del SCD hemos utilizado conversores PCF8591 con 4 conversores ADC y 1 DAC a través del bus I2C, y expansores PCF8574 que añaden al sistema 8 puertos de E/S digitales adicionales.

Se han probado diferentes unidades de control que tienen un requisito común: ser soluciones empotradas comerciales en los que se utiliza Java como lenguaje de programación principal para su control. Entre las empotradas probados podemos destacar el Javelin de Parallax Inc, placas de desarrollo basadas en procesadores Java aJile aj100 o aj80, como los JPro de Quest Technologies y JStik o JStamp de Systronix Inc, la placa de desarrollo para el modulo Snap de Imsys Tech, las placas de la compañía Snijder Micro System, SK2 y SC210. En el trabajo [2] se ha hecho un análisis pormenorizado de las características de dichas tarjetas. El empotrado que mejor se ajustaba a estos requerimientos es la placa SmartControl SC210 de Snijder, ya que además de contener una pantalla LCD táctil, disponía de suficientes puertos de entrada/salida digitales y analógicos, soporte Ethernet y la posibilidad de añadir más puertos de E/S a través del bus digital I2C.

La implementación del software de control se ha realizado utilizando el entorno de desarrollo Java que se distribuye con cada placa, sobre el que se añade la librería HAJASE, que proporciona la abstracción y adaptabilidad necesaria para ser utilizada con cualquiera de los empotrados probados utilizando una misma API común [2].

4 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Al igual que el hardware, los componentes software se han diseñado buscando la reutilización de todo el sistema o partes de él, facilitando la adaptación y extensión del sistema a nuevas funcionalidades o a la incorporación de nuevos dispositivos al sistema. Para garantizar el cumplimiento de los requerimientos del sistema, así como obtener componentes que se ajusten a dichos requerimientos, hemos seguido para el desarrollo del software una metodología orientada a objetos basada en UML.

A diferencia de lo que suele ser habitual en otras propuestas domóticas, el sistema SCD se ha

implementado completamente en Java, incluido los programas de control del Subsistema de Control. Para poner en funcionamiento el sistema necesitamos que cada componente tenga un entorno de ejecución o sistema operativo sobre el que se pueda añadir una máquina virtual Java (JVM) necesaria para la ejecución de las aplicaciones Java. Existen varias implementaciones de la JVM que ocupan desde pocos Kilobytes hasta más de 100 Mb de RAM que dependen de las características de la plataforma hardware, del sistema operativo subyacente y del conjunto de librerías que se implementan. Cada librería compuesta por una serie de paquetes y clases define una especificación Java que proporciona soporte para un campo de aplicación o tecnología concreta. Así, por ejemplo, la especificación JSR-82 [4] hace referencia a la librería Java que proporciona soporte Bluetooth a la JVM que lo implementa. No siempre se distribuye la JVM con todas las librerías que se pueden programar con ella, y se incluyen como librerías adicionales. También es posible encontrar JVMs dentro de lo que se ha denominado Edición; en este caso se incluye junto a la JVM un conjunto de librerías para un mercado concreto. Por ejemplo, la J2ME es una edición específica para los dispositivos portátiles y empujados, mientras que la J2SE es la edición estándar para la programación de equipos de sobremesa.

En el sistema SCD se utilizan una gran variedad de JVMs dada la variedad utilizada tanto en el acceso al SCD como para el control de los dispositivos domóticos. En resumen, utilizamos:

a) J2SE para la ejecución del Servidor Domótico del SCD y las máquinas clientes que acceden al Servidor Domótico desde cualquier punto de internet. Incluye el soporte de sockets para implementar las comunicaciones cableadas por Ethernet así como comunicaciones inalámbricas Wifi que utilizan TCP/IP y, por otra parte, se utiliza Swing para implementar el interfaz de usuario.

b) J2ME/CLDC/MIDP para la ejecución de los clientes en dispositivos móviles como teléfonos móviles y PDAs, e incluso para la ejecución de algunas unidades de control. Nos podemos encontrar diferencias significativas entre unos dispositivos y otros con respecto a las versiones de la configuración y perfil. Para poder soportar la conectividad a través de Wifi es necesario que la máquina virtual soporte las conexiones a través de Sockets, disponible desde la versión 2.0 de MIDP. Algunos dispositivos como, por ejemplo, los móviles, incluyen de fábrica una máquina virtual J2ME. Sin embargo, en dispositivos móviles como los PDAs, es necesario instalar previamente una máquina virtual J2ME para poder ejecutar las aplicaciones. Para garantizar que el dispositivo móvil pueda acceder al SCD se requiere

que incluya una JVM que incluya las librerías más recientes CLDC 1.1 y MIDP 2.0.

c) JSR-82. Para disponer soporte Bluetooth es necesario que la JVM incorpore una implementación de la librería, como ocurre en los teléfonos móviles, o bien se incorporen como una librería adicional en el caso de PDAs u ordenadores y portátiles.

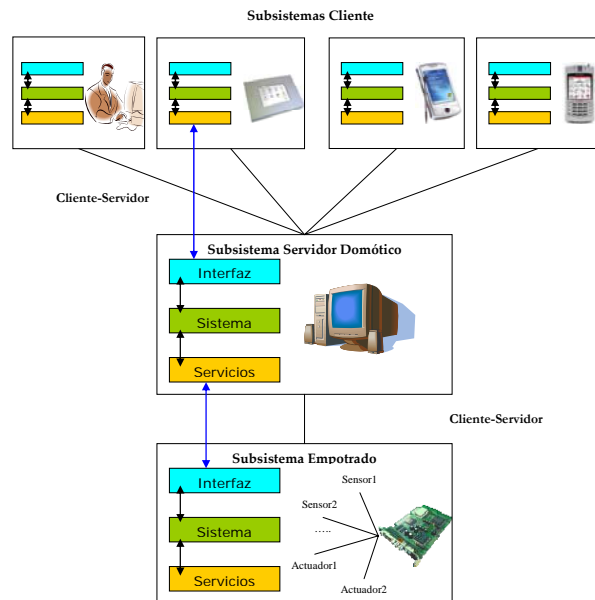


Figura 4. Arquitectura del sistema.

4.1 ARQUITECTURA DEL SISTEMA

La arquitectura utilizada para el diseño del software es una combinación de una arquitectura cliente-servidor y una arquitectura multicapa en tres capas como se puede ver en la figura 4. La separación que propugna la arquitectura cliente-servidor nos permite separar totalmente el subsistema cliente respecto del subsistema servidor, lo que permite desacoplar dichos subsistemas, mejorar la cohesión de cada subsistema, facilitar la reconfiguración y la realización de pruebas de los subsistemas, y permitir la reutilización y/o sustitución de los subsistemas.

Además de la arquitectura cliente-servidor, hemos utilizado una arquitectura en tres capas para el diseño de cada uno de los subsistemas señalados anteriormente, que son: interfaz, dominio de aplicación y servicios. Cada capa a su vez se va a descomponer en nuevos subsistemas, que denominamos subsistemas estructurales internos para diferenciarlos de los anteriores y que tendrán una responsabilidad más delimitada y restringida en función de la capa en la que se encuentra. Se ha escogido esta arquitectura para el diseño de cada subsistema entre otras razones por las ventajas que aporta en cuanto a la reutilización de los

componentes del sistema, su adaptabilidad de modo que cualquier cambio en el componente no afecta al resto de componentes o subsistemas del sistema, y la capacidad de sustitución de un subsistema por otro de forma transparente siempre que se preserve el mismo interfaz del subsistema.

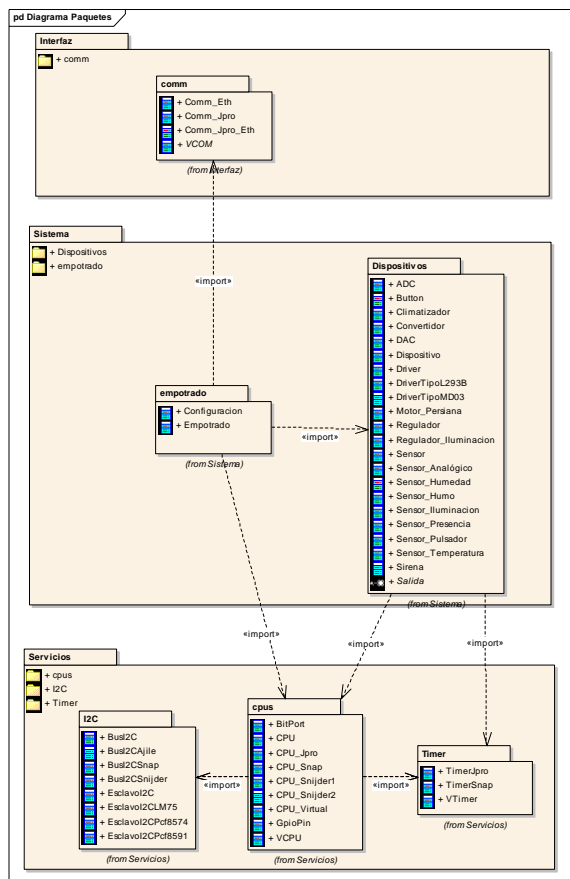


Figura 5. Diagrama de paquetes del Subsistema de Control.

Esta forma de estructurar el sistema nos facilita la implementación tanto del SCD como de los diferentes subsistemas que se ejecutan paralelamente en el SCD. De modo que es factible implementar una nueva aplicación cliente del Subsistema Clientes que puede acceder al SCD utilizando un nuevo protocolo de comunicaciones sin más que adaptar la capa de servicios para que pueda incluir el nuevo protocolo de comunicaciones reutilizando el resto de capas de la aplicación cliente. Esto posibilita una simplificación en el desarrollo de nuevos componentes del sistema, o al menos la reducción en el tiempo de desarrollo de dichos componentes.

El modelo arquitectónico propuesto proporciona la estructura en la que se sustenta cada componente del sistema. En la figura 5 se muestra, por ejemplo, el diagrama de paquetes del Subsistema de Control del SCD de acuerdo a dicho modelo arquitectónico. Se puede observar tanto los paquetes como las clases

contenidas en cada paquete que conforman el subsistema, así como las relaciones de dependencia entre los diferentes paquetes.

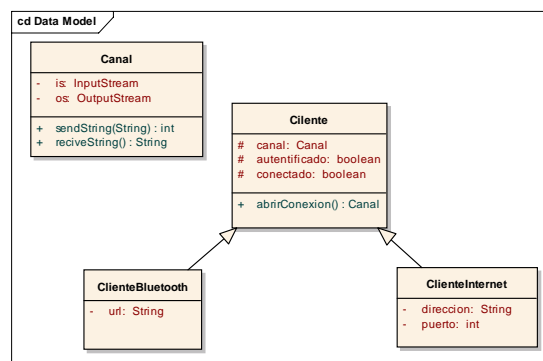


Figura 6. Abstracción de la comunicación del cliente.

4.2 AGREGACIÓN DE UN PROTOCOLO DE COMUNICACION

Para mostrar como funciona el mecanismo de extensión del sistema, vamos a mostrar un ejemplo de su utilización para el caso de los protocolos de comunicaciones. La forma en como se ha diseñado el sistema, permite modificar el protocolo de comunicaciones en el cliente sin que afecte en exceso al resto del sistema para lo cual hay que abstraer en la mayor medida de lo posible los detalles de cada tecnología de comunicación manteniendo inalterada la lógica del programa.

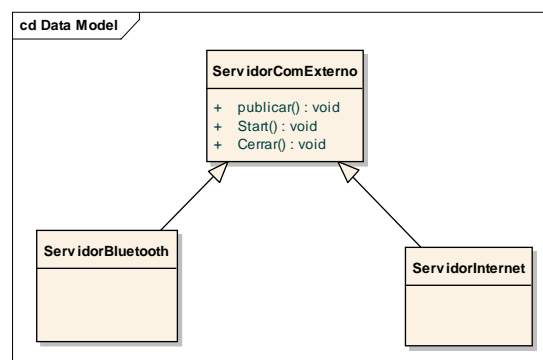


Figura 7. Abstracción de la comunicación del servidor.

En la figura 6 se muestra un ejemplo sencillo de uno de los mecanismos empleados para la extensión del sistema basado en la herencia, aunque el SCD emplea distintos patrones de diseño como el patrón factoría, el patrón adaptador, etc. [1], que incluyen la delegación además del mecanismo de herencia. En este caso, los protocolos de comunicaciones se implementan a partir de especializaciones de una clase base utilizando el mecanismo de herencia. Para incluir un nuevo Cliente basado en una nueva tecnología de comunicación solamente debemos de

heredar de la clase Cliente y redefinir el método abstracto `abrirConexión()` donde se incluyen los pasos específicos para abrir un canal de comunicaciones (clase Canal) en la nueva tecnología de comunicación.

En la parte del Servidor (figura 7) se ha adoptado un diseño parecido aunque la incorporación de una nueva tecnología de comunicaciones es algo más compleja ya que hay que redefinir un número mayor de métodos. Además el servidor debe gestionar la conexión concurrente de múltiples peticiones de cliente para lo cual se necesita implementar un hilo de ejecución por cada sesión.

4.3 SIMULADOR DE LA UNIDAD DE CONTROL

Para probar el software desarrollado y depurar las aplicaciones que utilizan la unidad de control se ha desarrollado un simulador de la unidad de control, denominado empotrado virtual, para poder trabajar con él sin necesidad de tenerlo conectado físicamente. Además de la independencia con respecto al sistema real, proporciona la posibilidad de modificar las condiciones del entorno sin necesidad de disponer del modelo físico real.

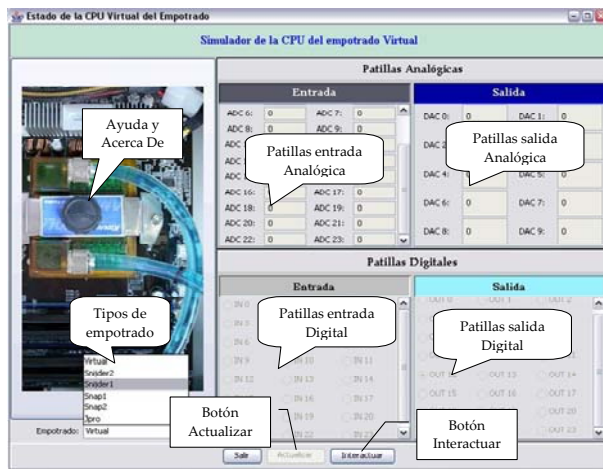


Figura 8. Pantalla del simulador del empotrado virtual.

El simulador es un programa independiente que puede ejecutarse en cualquier ordenador, y que puede ser configurado con cualquiera de los empotrados reales utilizados durante el desarrollo, permitiendo así trabajar con la unidad de control de forma virtual.

Además es posible modificar en tiempo de ejecución la configuración del hardware que queremos simular, así como modificar el estado de cualquiera de los puertos de entrada/salida digital o analógico que puede tener la unidad de control.

La librería HAJASE proporciona una implementación adicional para la unidad de control genérica, de modo que el desarrollo del software del SCD se puede realizar de forma desacoplada respecto de la unidad de control.

5 CONCLUSIONES Y TRABAJO FUTURO

Se ha desarrollado un sistema para el control domótico completo de una vivienda tanto a nivel hardware como a nivel software, buscando construir un sistema abierto, flexible, modularizable, extensible, interoperable, escalable para lo cual hemos utilizados tecnologías basadas en Java. El resultado de este trabajo demuestra que es posible utilizar Java como lenguaje de programación para el desarrollo de todo el sistema, incluso la programación de los controladores de los dispositivos con requerimientos temporales, aunque para ello tenemos que sacrificar alguna de las facilidades que ofrece Java a los desarrolladores.

El siguiente paso, consistiría en aplicar el sistema desarrollado a un entorno domótico real para algunas de las redes domóticas como X10, Lonworks, EIB, etc., utilizando dispositivos domóticos reales. El desarrollo se haría manteniendo la arquitectura mostrada en el artículo, pero extendiendo el sistema añadiendo nuevos componentes específicos para cada protocolo de comunicación nuevo y para cada dispositivo domótico nuevo utilizando la jerarquía de clases definida para virtualizar el uso de cada uno de los dispositivos domóticos.

Referencias

- [1] Kuchana, P, (2004) "Software Architecture Design Patterns in Java", Auerbach Publications.
- [2] Montes J.M y Viudez, (2005) "Sistema de Control Domótico", Proyecto Fin de Carrera. Universidad de Granada.
- [3] Romero C., Vázquez F., Lozano C. (2005) "Domotica e Inmótica", Ra-Ma.
- [4] Sun Microsystems, <http://java.sun.com>

APROXIMACIÓN AL RECONOCIMIENTO VISUAL AUTOMÁTICO DE LA LENGUA DE SIGNOS

Isaac García Incertis
Fundación CARTIF
isagar@cartif.es

Jaime Gómez García-Bermejo
Universidad de Valladolid
jaigom@eis.uva.es

Eduardo Zalama Casanova
Universidad de Valladolid
ezalama@eis.uva.es

Resumen

La comunicación hombre-máquina es un campo en crecimiento en los últimos años, pero la máquina también puede servir para comunicar a los hombres entre sí. Ambos aspectos pueden quedar cubiertos simultáneamente cuando el usuario es sordo y su lenguaje natural es la Lengua de Signos. En éste artículo se propone un método para el reconocimiento del alfabeto dactilológico de la Lengua de Signos Española mediante Visión por Computador, que emplea la combinación de una serie de normas, que miden la distancia entre el signo presentado y los modelos almacenados en una base de datos (diccionario). Dicho método es fiable y computacionalmente ligero, lo que permite un reconocimiento de los signos en tiempo real y con un equipo de bajo presupuesto.

Palabras Clave: Visión por computador (CV), reconocimiento de gestos (GR), interacción hombre máquina (HMI).

1 INTRODUCCIÓN

El reconocimiento de gestos con la mano se está convirtiendo en una corriente preferente en el reconocimiento de patrones [6, 17], dado el creciente número de aplicaciones en el campo de la interacción hombre-máquina (p.e. domótica, asistencia a personas dependientes, dispositivo de entrada en interfaces gráficas, manipulación en entornos de realidad aumentada [14]). En este artículo se trata el reconocimiento de gestos manuales en la comunicación con las personas sordas mediante técnicas de visión por computador [3, 7, 13]. La interpretación de la Lengua de Signos (LS) en general es un todo un desafío dada su indudable

proyección social y sus problemas intrínsecos: (i) la naturaleza multicanal de la comunicación (la postura de las manos y su orientación, el movimiento y su velocidad, su posición relativa al “espacio de signado” y otras partes del cuerpo, la expresión del rostro que acompaña a un signo determinado,...); (ii) la variabilidad entre personas al ejecutar el signado (faceta bien conocida en los reconocedores de voz); (iii) y la presencia de elementos ajenos a la comunicación (ropas, fondos cambiantes, oclusiones). En este trabajo nos centraremos en el reconocimiento de los signos estáticos que representan las letras de la Lengua de Signos Española (LSE).

El reconocimiento de signos con la mano se ha abordado en [10, 14], pero sólo para contar el número de dedos estirados, independientemente de cuáles sean. También se ha investigado en [7], pero no con técnicas de visión sino mediante el uso de ciberguantes sensorizados, que informan mediante sensores de la postura adoptada por la mano, simplificando el reconocimiento pero encareciendo enormemente el equipo necesario. En [1] se propone un enfoque más general para elementos articulados que se puede adaptar al reconocimiento de la LSE, pero requiere resolver un costoso problema de optimización para cada imagen. En [13] se propone el reconocimiento de gestos mediante momentos, pero dicha solución sólo es apropiada para discriminar un limitado conjunto de signos, al igual que los invariantes de los momentos, los descriptores simples de formas y los histogramas de chain-codes expuestos en [5, 11, 15] respectivamente. En [4, 16] se propone el uso de histogramas geométricos y en [8, 9] el uso de curvaturas nulas y su caracterización. En [2] se aborda el reconocimiento mediante modelos de partículas. Finalmente en [12] y [3] se aborda el problema en 3D, mediante sistemas multicamerales y modelos con muchos grados de

libertad, exigiendo además un seguimiento continuo de la mano.

En este artículo se presenta un método fácil de implementar, poco exigente en recursos de cálculo, que no necesita imágenes de gran calidad (basta una simple webcam en color) y pese a todo, capaz de discriminar la mayoría de los signos alfabéticos de la LSE.

En la sección 2 se describe el método propuesto. Los resultados experimentales se muestran en la sección 3 y las conclusiones en la sección 4.

2 DESCRIPCIÓN DEL MÉTODO

A grandes rasgos el método opera en la forma siguiente: En primer lugar se segmenta la región de la mano y su contorno, a continuación se muestrea el contorno según arcos de longitud constante y los puntos muestreados se comparan con los perfiles almacenados previamente en nuestro diccionario. Dicha comparación se realiza teniendo en cuenta cuatro métricas diferentes, lo cual proporciona la robustez adecuada al método.

2.1 SEGMENTACIÓN DE LA MANO

La imagen RGB que proporciona la cámara se convierte al espacio de color HSV (tono, saturación, brillo), dado que esta representación es más adecuada para la detección del color y la segmentación en el plano HS muestra una gran robustez frente a las variaciones de iluminación. La transformación se ha

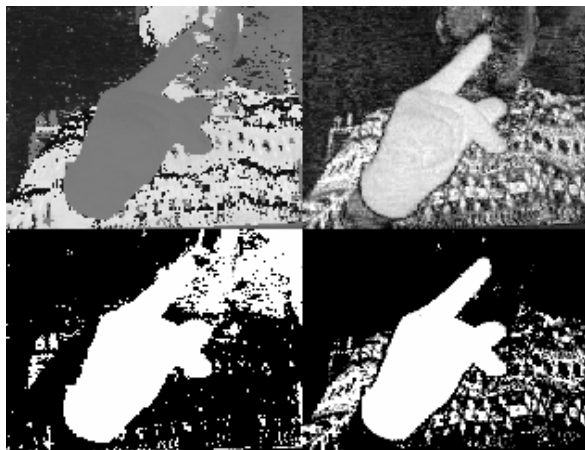


Figura 1. Tono (izquierda) y saturación (derecha) en niveles de gris (arriba) y filtrados (abajo).

adaptado para que el tono (hue) de interés esté en torno a los 180° (127 en la representación en un byte) para evitar los problemas del paso por cero típicos de las magnitudes angulares, como puede ser el círculo de tonalidad del color (figura 1). Una vez aplicado el filtro en el plano HS se realizan algunas operaciones



Figura 2. Segmentación de la mano

de morfología matemática para eliminar al menos parte del ruido (figura 2) y se extrae el contorno de las regiones con un área y perímetro suficiente para considerar que es una mano. Dichos contornos son el punto de partida para la siguiente etapa en el proceso de reconocimiento. (El método no permite discriminar por ejemplo una 'F' de una 'T', al tener la misma silueta.)

2.2 MUESTREO DEL CONTORNO

Una vez obtenido el contorno de la mano se toman muestras equiespaciadas, para obtener un conjunto de puntos lo suficientemente grande para no perder detalles significativos y lo bastante pequeño para no sobrecargar los cálculos innecesariamente. Por ejemplo en la figura 3 se muestran 100 puntos muestreados del contorno de la letra 'K'.



Figura 3. Puntos muestreados y contornos internos.

Los puntos muestreados se convierten a coordenadas polares tomando como origen el centro de gravedad de la región. El resultado se puede ver en la figura 4 donde destacan, de izquierda a derecha, los picos correspondientes a los dedos índice, pulgar y corazón, y la meseta de la muñeca.

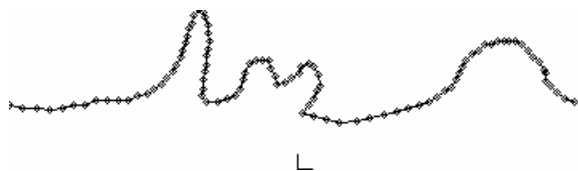


Figura 4. Diagrama polar (radio vs. ángulo)

El radio es escalado para obtener invarianza frente al tamaño aparente de la mano, que varía con la distancia del signante a la cámara. Ello permite identificar gestos donde la mano ocupa casi toda la imagen (como los mostrados en éste artículo) y gestos donde la imagen abarca todo el tronco y cabeza del signante (forma habitual de uso del sistema).

2.3 COMPARACIÓN CON LOS MODELOS

2.3.1 Generación del diccionario de modelos

El modelo de cada signo se obtiene estableciendo una banda de confianza entorno al diagrama de radios de los puntos muestreados (figura 5). Para establecer dicha banda se siguen dos pasos, el primero es aumentar y disminuir el valor correspondiente a radio de cada punto en un porcentaje fijo, lo que nos da una buena aproximación para contornos con variaciones suaves del radio, pero con mal comportamiento cuando se producen variaciones bruscas (pendientes altas) de éste, como las que se pueden ver en la figura 5.

Para evitar éste problema se procede a sustituir cada punto de la superior (respectivamente inferior) de la banda por el máximo (r. mínimo) que se produce en una ventana centrada en el punto a considerar. Este proceso es formalmente idéntico a aplicar una dilatación (r. erosión) de morfología matemática unidimensional al extremo superior (r. inferior) de la banda, y aproxima eficientemente el hecho de considerar la anchura de la banda relacionada linealmente con la varianza del radio en la ventana considerada.

El conjunto de ambos pasos es equivalente también a realizar sobre la curva del radio una dilatación, esta vez bidimensional, con un elemento estructurante rectangular centrado, cuya altura varía de forma proporcional a la magnitud del radio en cada punto. La banda de confianza (cuadrados) obtenida se puede ver en la figura 6 junto a la curva del radio (puntos) en los puntos muestreados para otra representación de la misma letra.



Figura 6. Radio y banda de confianza frente a longitud de arco

2.3.2 Búsqueda en el diccionario

La sucesión de puntos (su radio escalado) se compara con cada uno de los modelos almacenados en el diccionario. La comparación es mucho más selectiva si combinamos mediante un sistema de votación los resultados obtenidos por cada una de las cuatro métricas aplicadas.

Primero se comprueba para cada punto si está dentro de la banda de confianza. De ser así éste punto no computa en el cálculo de la distancia, y en caso contrario se mide la distancia (en vertical) del punto a la banda. Las distancias medidas entre el signo a identificar y cada uno de los modelos almacenados son:

- L_0 : número de puntos que están fuera de la banda.
- L_1 : suma de las distancias de los puntos a la banda.
- L_2 : suma de los cuadrados de dichas distancias.
- L_{inf} : máximo de dichas distancias.

Una vez calculadas dichas distancias para cada una de las cuatro métricas se comprueba si el cociente de distancias del modelo más próximo y el siguiente clasificado para esa misma métrica es sensiblemente pequeño y en caso afirmativo se considera que para esa métrica ése es el modelo correspondiente. Si el cociente no es lo bastante pequeño (1 en el caso extremo) significa que con esa métrica no hemos encontrado un modelo marcadamente similar al gesto que pretendemos identificar.

Si 3 ó 4 de las métricas coinciden en su veredicto se considera que ha habido una identificación positiva del signo. En la figura 7 se puede comprobar la identificación positiva de la letra 'K' mediante los cuatro criterios.

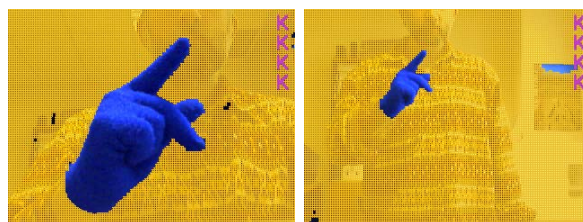


Figura 7. Reconocimiento de signos en imágenes próximas y lejanas

3 RESULTADOS EXPERIMENTALES

El método descrito en la sección 2 se ha implementado en un ordenador dotado con un microprocesador Pentium IV a 3GHz y una cámara Logitech. El SO empleado ha sido Windows 2000. La captura de imágenes se ha realizado con una resolución de 320x240 píxeles y haciendo uso de la librerías OpenCV de Intel se realiza el procesamiento a una tasa de unas 12 fps, si bien más del 80% del tiempo es utilizado en la adquisición de imagen y la conversión de formato que realizan dichas librerías.

Tabla 1: Tasas de reconocimiento

A	99.3%	G	85.0%	M	91.7%	R	93.3%
B	99.7%	H	55.3%	N	95.7%	S	62.0%
C	96.3%	I	97.7%	O	78.3%	U	97.3%
CH	95.7%	K	86.0%	P	99.0%	W	98.7%
E	99.3%	L	98.0%	Q	92.7%		

Para obtener las tasas de reconocimiento que se muestran en la tabla uno se ha procedido a signar cada letra repetidas veces, sin que el signante tenga realimentación acerca del reconocimiento realizado por el sistema y en condiciones variadas de iluminación y posición relativa a la cámara. Las tasas de reconocimiento son extraordinariamente altas, superando en algunos casos el 99%. Los casos no contemplados corresponden bien a letras cuyo signado coincide con otras pero añadiendo movimiento (J, Y y Z con la I, LL con la L, Ñ con la N, RR con la R, V con la U) o bien a letras cuya silueta se puede confundir fácilmente con otras (o son idénticas, como F y T), por lo que sería necesario un procesamiento posterior para discriminar entre éstas. No obstante algunas de ellas se han contemplado en el experimento con mejor (C y E, figuras 8 y 9) o peor (O y S) resultado.



Figura 8. Letra C en la LSE

Es conveniente recordar que el experimento se ha llevado a cabo sin que el signante tenga

conocimiento de si el sistema a reconocido el signo o no, sabiendo únicamente si la mano está en cuadro en la imagen. Es de suponer que en una aplicación práctica se puede informar al signante sobre el reconocimiento para que aquél modifique el signado en línea, en caso de no ser reconocido por el sistema. Algunas pruebas realizadas en este sentido han mostrado una notable mejora (cuando esto era posible) de las tasas de reconocimiento del sistema.



Figura 9. Letra E en la LSE

4 CONCLUSIONES

Se ha presentado un método de fácil implementación, y bajo costo computacional que afronta con éxito la correcta identificación de los signos propuestos con una tasa de imágenes por segundo adecuada para su propósito.

En el futuro se prevé analizar la influencia de un aumento masivo del diccionario de gestos, si bien cabe esperar que tal circunstancia sea fácilmente superable dado que el presente método permite la ampliación de forma sencilla tanto en horizontal (agregando nuevas métricas o sustituyendo algunas de las actuales) como en vertical. En los casos dudosos se podrían establecer criterios de discriminación adicionales, basados en información no utilizada actualmente. Por ejemplo en el caso de la F y la T, se podría analizar la dirección dominante de contornos internos o de los gradientes en la zona donde se cruzan los dedos corazón y pulgar según se muestra en las figuras 10 y 11. En todo caso la incorporación de diccionarios de más alto nivel (sintácticos y semánticos) resulta el camino natural mejorar la tasa de reconocimiento.



Figura 10. Signos para las letras F y T

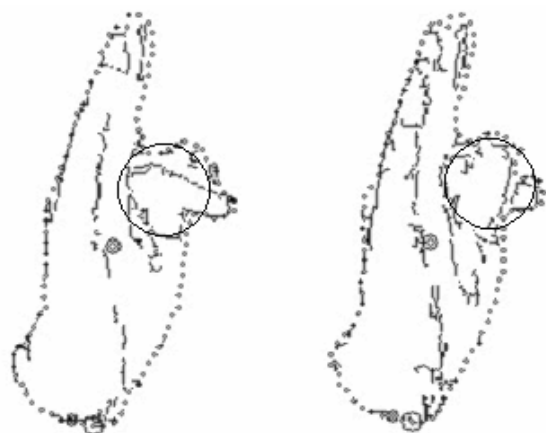


Figura 11. Procedimiento adicional para diferenciar las letras F y T

Agradecimientos

La investigación realizada en la Fundación CARTIF que ha conducido al presente artículo ha sido parcialmente financiada por el programa Profit (proyecto N° FIT-350300-2004-22, ejecutado en coordinación con las empresas Telefónica I+D y RedHada).

Referencias

- [1] Black, M.J.; Jepson, A.D. (1996) "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation", *Fourth European Conference on Computer Vision*, pp. 329-342.
- [2] Bretzner, Lars; Laptev, Ivan; Lindeberg, Tony (2002) "Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering", *Proc. Face and Gesture*, Washington D.C., pp. 423-428.
- [3] Holden, Eun-Jung; Owens, Robyn (2001) "Visual Sign Language Recognition", *Lecture*

Notes in Computer Science Vol. 2032, pp. 270-274.

- [4] Huet, Benoit; Hancock, Edwin R. (1998) "Relational histograms for shape indexing", *Proc. of ICCV*, pp. 563-569.
- [5] Iivarinen, J.; Visa, A. (1996) "Shape recognition of irregular objects", Ed. D.P. Casasent, *"Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904"*, pp. 25-32.
- [6] Jain, Anil K.; Duin, Robert P. W.; Mao, Jianchang, (2000) "Statistical Pattern Recognition: A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4-37.
- [7] Kadous, Mohammed Waleed (1998) "Auslan sign recognition using computers and gloves", *Deaf Studies Research Symposium*.
- [8] Leymarie, F.; Levine, M. D. (1988) "Curvature morphology", *Tech. Rep. TR-CIM-88-26*, Computer Vision and Robotics Laboratory, McGill University, Montreal, Quebec, Canada.
- [9] Mokhtarian, Farzin; Abbasi, Sadegh; Kittler, Josef (1996) "Efficient and robust retrieval by shape content through curvature scale space", *Proc. International Workshop on Image Databases and MultiMedia Search*, Amsterdam, The Netherlands, pp. 35-42.
- [10] New, Joshua R. (2002) "A Method for Hand Gesture Recognition", *Proceedings of the ACM Mid-Southeast Chapter Fall Conference*, Gatlinburg, Tennessee, USA.
- [11] Peura, Markus; Iivarinen, Jukka (1997) "Efficiency of Simple Shape Descriptors", *3rd International Workshop on Visual Form*, Capri, Italy.
- [12] Rehg, James M.; Kanade, Takeo (1994) "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking", *Third European Conference on Computer Vision*, pp. 35-46.
- [13] Starner, T.; Pentland, A. (1995) "Visual recognition of american sign language using hidden markov models", *International Workshop on Automatic Face and Gesture Recognition*, pp. 189-194.

- [14] Störking, Moritz; Moeslund, Thomas B.; Liu, Yong; Granum, Erik (2004) "Computer vision-based gesture recognition for an augmented reality interface", *4th IASTED International Conference on Visualization Imaging and Image Processing*, pp. 766-771.
- [15] Suk, Tomás ; Flusser, Jan (2003) "Combined Blur and Affine Moment Invariants and their use in Pattern Recognition", *The Journal of the Pattern Recognition Society*, Vol. 36, pp. 2895-2907.
- [16] Thacker, N.; Riocreux, P.; Yates, R. (1995) "Assessing the completeness properties of pairwise geometric histograms", *Image and Vision Computing*, 13(5), pp. 423-429.
- [17] Wu, Ying; Huang, Thomas S. (1999) "Vision-Based Gesture Recognition: A Review", *Lecture Notes in Computer Science Vol. 1739*, pp. 103-114.

UNA METODOLOGÍA DE BAJO COSTE COMPUTACIONAL PARA EL CONTROL PREDICTIVO DE SISTEMAS NO LINEALES MEDIANTE REDES NEURONALES

Eloy Irigoyen Gordo

Dept. Ingeniería de Sistemas y Automática (UPV-EHU), eloy.irigoyen@ehu.es

Resumen

En este trabajo se desea realizar de forma satisfactoria el control de sistemas no lineales, mediante una estrategia de Control Predictivo y utilizando herramientas de Computación Inteligente, como son las Redes Neuronales. Lo novedoso de este trabajo son los métodos que se emplean para reducir de forma notable el gasto computacional, que es de especial relevancia en aplicaciones donde el mecanismo de control se adapta en tiempo real. Estos métodos pasan por la eliminación de recursividades en el lazo de control, el aumento de la velocidad de entrenamiento de la Redes Neuronales y la identificación de los sistemas no lineales mediante un emulador neuronal, que contiene implícitamente el modelo del sistema a controlar.

Palabras Clave: Redes Neuronales, Control Predictivo, Sistemas No Lineales, Reducción de Gasto Computacional

1 INTRODUCCIÓN

Son múltiples las propuestas para resolver los problemas de control existentes en el ámbito de la estrategia del Control Predictivo (CP). Acerca del control de Sistemas Lineales se han desarrollado amplios y exhaustivos trabajos, y reflejo de todo ello es la cantidad innumerable de artículos existentes hoy en día. Por otro lado, en estos momentos se están haciendo grandes esfuerzos en estudiar el problema del control de Sistemas No Lineales. Las propuestas aparecidas en los últimos años abarcan un amplio espectro. En todas ellas aparecen unos denominadores comunes como son: el uso implícito o explícito de un modelo del sistema a controlar, la predicción del valor de las variables futuras del sistema de control en un horizonte establecido previamente, la minimización de ciertas funciones objetivo o de costo, el cálculo consecuente de una secuencia de control y la aplicación de una estrategia de horizonte móvil [1]. La mayoría de estas propuestas señalan la importancia de un bajo gasto

computacional, aunque el peso de sus estudios recae en la creación de una nueva metodología de control predictivo.

En muchos de los estudios de los problemas de control de sistemas no lineales mediante la teoría de CP se desarrollan diferentes ideas, en donde tiene especial relevancia el diseño de la función de coste, que posteriormente se incorpora a un proceso de optimización para la búsqueda de un valor mínimo satisfactorio. En este sentido se pueden encontrar propuestas donde se plantea una reducción de la complejidad de dicha expresión eliminando o no considerando ciertos términos [4,5]. Estas propuestas pueden derivar en una notable mejora en la reducción del gasto computacional, ofreciendo resultados adecuados en el control de sistemas no lineales relativamente complejos [6,7].

En el campo de la Computación Inteligente, concretamente en la línea de las Redes Neuronales (RNs), desde hace varias décadas han ido apareciendo trabajos destinados a la solución de problemas de la teoría de control [9,10]. Estas han sido utilizadas indistintamente tanto en el desarrollo y sintonización de todo tipo de controladores, como en el proceso de identificación de sistemas lineales y no lineales [2]. En trabajos más recientes han surgido propuestas donde se integran las RNs en el sistema de control para ambos dispositivos, controlador e identificador. En un nuevo paso este planteamiento se está llevado a los sistemas no lineales, donde la problemática de control es mucho más compleja de resolver [3].

Los procesos de entrenamiento de las RNs pueden llegar a ser muy costosos en el tiempo. Los procesos de ajuste de sus parámetros, pesos y bias, pueden llegar a necesitar una gran cantidad de iteraciones para alcanzar un valor mínimo previamente establecido. En la mejora de estos procesos de entrenamiento han aparecido múltiples trabajos a lo largo de los últimos años [8,12]. Esta cuestión es de vital importancia si se traslada a un sistema de control, donde tanto el controlador como el identificador son RNs, y además, el ajuste de sus parámetros se debe hacer en tiempo real. Esta es la

situación que se produce en los planteamientos de CP. Por todo ello, recoger dichas mejoras e integrarlas en este tipo de aplicaciones redundará en una mejora del sistema de control, donde las exigencias temporales pueden desbordar a los planteamientos más novedosos aparecidos recientemente.

2 OBJETIVOS

En este trabajo se estudia la reducción de los tiempos de cómputo en un sistema de control que adapta sus dispositivos en tiempo real, empleando una serie de métodos, ya estudiados y contrastados, integrándolos en los diferentes componentes que existen en dicho sistema de control. Además, en este sistema de control se utilizarán las RNs para la creación de un controlador no lineal [6], y de un identificador del sistema a controlar [11]. La dificultad de este trabajo no radica en la creación de una estructura del sistema de control compleja, tal y como se puede observar en la figura 1. Su aportación viene dada por la incorporación de nuevas metodologías en todas las etapas del proceso de control, tanto en la sintonización de los parámetros del controlador neuronal, como en la selección de una función de coste suficientemente sencilla para el proceso de optimización, como de la configuración del identificador neuronal. Todo ello para alcanzar el objetivo principal: reducir al máximo el gasto computacional y mejorar las prestaciones en el tiempo de estos dispositivos.

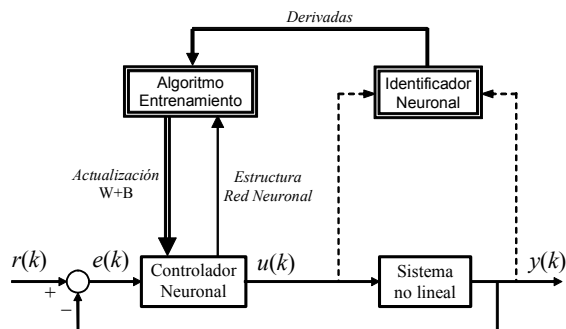


Figura 1: Esquema de control

Todo ello se realizará sobre sistemas no lineales discretos en el tiempo, algunos de los cuales presentan un comportamiento inestable en ciertos puntos de trabajo, haciendo más difícil su control. Un ejemplo de este tipo de sistemas se puede observar en la fórmula 1 (SNL3) [6].

$$y_{k+1} = K_1 \cdot u_k^3 + \frac{K_2 \cdot y_k - K_3 \cdot y_{k-1}}{1 + K_4 \cdot y_k^2} \quad (1)$$

$K_1 = 0.3; K_2 = 2.4; K_3 = 1.1; K_4 = 0.8$

Estos sistemas se estudiarán previamente para determinar la complejidad de su control en función de su comportamiento en determinados rangos de trabajo. Una representación de dicha característica se muestra en la figura 2.

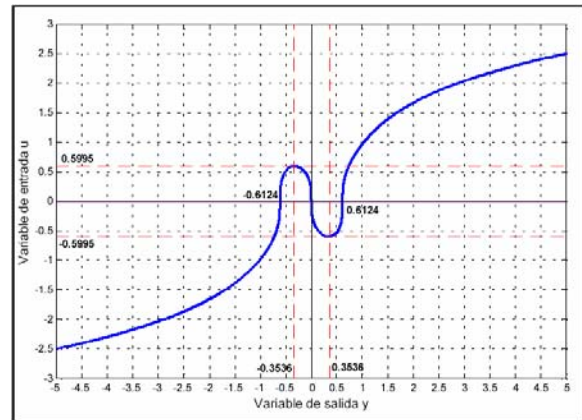


Figura 2: Puntos de trabajo del sistema SNL3

3 CONTROLADOR NEURONAL

El esquema de control sobre el que se va a desarrollar el trabajo es un esquema de lazo cerrado estándar, como el presentado en la figura 1, donde una RN de carácter no lineal es utilizada como controlador. Su estructura está basada en un controlador PID generalizado, tal y como se puede observar en la figura 3.

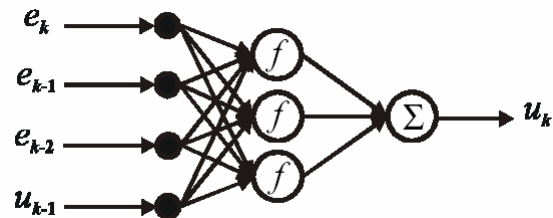


Figura 3: Controlador neuronal no lineal

El carácter no lineal del controlador viene dado por las funciones de salida de las neuronas de la capa oculta que son del tipo sigmoideal. Además, este tipo de dispositivos permiten limitar de forma sencilla el rango de actuación de la señal de control, al cambiar la función lineal de la neurona de la capa de salida por otra de forma sigmoideal. Además, mediante una pequeña modificación en la estructura de la red se podría limitar igualmente las variaciones de la acción de control [6]. Esta circunstancia es muchas veces contemplada dentro de la estrategia de control, al incluir un término nuevo en la función de coste para su penalización. La eliminación de este término puede revertir en un ahorro del coste computacional al emplear una función de coste más sencilla.

Actualmente, existen diferentes propuestas para los procesos de entrenamiento de las RNs. De todos ellos se ha seleccionado un método de segundo orden mejorado, basado en el planteamiento de Levenberg-Marquardt, pero con un análisis de vecindad de parámetros de la RN para acelerar el proceso de entrenamiento [8,11]. Para el esquema de control que se presenta en la figura 1 ha sido necesario modificar tal método, dado que el conjunto de entrenamiento para la RN no se puede obtener de forma directa. Ello implica una modificación en su algoritmo, teniendo en cuenta las características dinámicas del sistema no lineal a controlar. En esta línea se debe tomar como error a minimizar el dado a la salida del sistema, lo que implica el desarrollo del cálculo de los términos derivados del error a través de dicho sistema. Esta es una problemática ya estudiada en varios trabajos [5,6] y que hace que ciertas expresiones deban desarrollarse tal y como se muestra en la fórmula 2.

$$\sum_{k=0}^K \frac{\partial E_{ek}}{\partial w_{lij}} = \sum_{k=0}^K \sum_{k'=0}^k \sum_{k''=0}^{k'} \frac{\partial E_{ek}}{\partial y_{k'}} \cdot \frac{\partial y_{k'}}{\partial u_{k''}} \cdot \frac{\partial u_{k''}}{\partial w_{lij}} \quad (2)$$

En este punto debe realizarse una nueva reflexión acerca del gasto computacional. El depender dichas expresiones de la dinámica del sistema no lineal genera un nuevo desarrollo de las mismas, como se puede observar en la fórmula 3.

$$\frac{\partial^+ y_{k'}}{\partial u_{k''}} = \sum_{i=1}^n \frac{\partial y_{k'}}{\partial y_{k'-i}} \cdot \frac{\partial^+ y_{k'-i}}{\partial u_{k''}} + \sum_{j=1}^m \frac{\partial y_{k'}}{\partial u_{k'-j}} \cdot \frac{\partial u_{k'-j}}{\partial u_{k''}} \quad (3)$$

Es aquí donde se aplican nuevas metodologías que no consideran ciertas recursividades en el sistema de control, reduciendo dichas expresiones y por tanto el gasto computacional del proceso de entrenamiento [6].

4 IDENTIFICADOR NEURONAL

Para el proceso de entrenamiento de la RN que realiza el papel de controlador es necesario conocer el Jacobiano del sistema no lineal, tal y como se puede observar en las fórmulas 2 y 3. El cálculo del mismo se realiza a través de una red neuronal no lineal, de la cual se pueden extraer las reacciones entre las variables de entrada y salida del sistema. Este planteamiento ha sido presentado previamente en diferentes trabajos [11]. En este trabajo se pretende optimizar en tiempo el entrenamiento de su RN, por medio de las propuestas apuntadas en el anterior capítulo. Además, se integrará en la estructura de control de tal manera que, cuando los requerimientos en tiempo y la buena evolución del entrenamiento del controlador neuronal lo permitan,

se resintonicen sus parámetros para disponer de un dispositivo que captura la dinámica del sistemas no lineal en tiempo real.

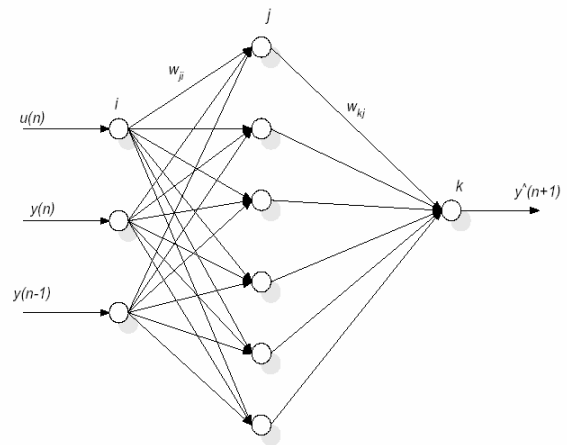


Figura 4: Emulador neuronal

La configuración básica de esta RN es como la que se muestra en la figura 4. Su tipo de estructura permite conocer los diferentes elementos del Jacobiano del sistema en base a la relación entre sus neuronas. A modo de ejemplo, se puede ver en la fórmula 4 la relación entre un elemento del Jacobiano $\frac{\partial \hat{y}(n+1)}{\partial u(n)}$, y los pesos w y salidas o de las neuronas de la RN.

$$\frac{\partial \hat{y}(n+1)}{\partial u(n)} = \sum_{j=1}^6 w_{1j} o_j (1 - o_j) w_{j1} \quad (4)$$

5 RESULTADOS

Los diferentes trabajos llevados a cabo hasta el momento han demostrado que esta metodología funciona de forma satisfactoria. Las pruebas realizadas con el controlador neuronal con el sistema SNL3 han ofrecido resultados como el mostrado en la figura 5. En ella se realiza el control teniendo en cuenta que la salida del sistema no lineal viene perturbada por una señal ruidosa, y además existe una incertidumbre del 5% en el modelo del sistema a controlar.

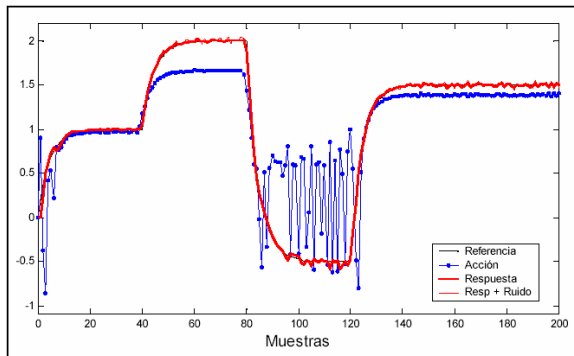


Figura 5. Control de SNL3 en base a modificaciones en los algoritmos de entrenamiento

Las pruebas donde se ha evaluado el comportamiento del entrenamiento de las RNs con los algoritmos modificados en base a los nuevos métodos mencionados anteriormente, han demostrado un de forma satisfactoria alcanzar las exigencias de control de los sistemas no lineales, tal y como se muestra en la figura 6.

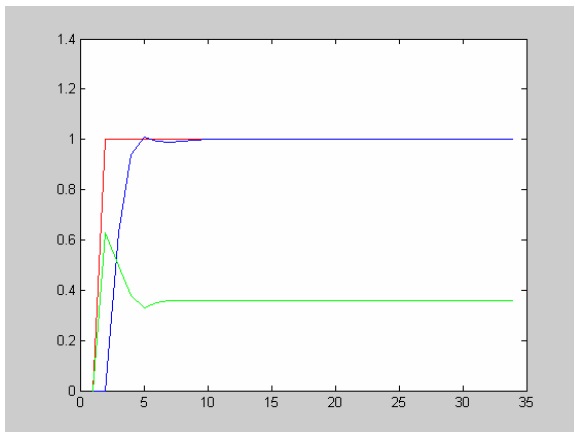


Figura 6. Control de SNL3 en base a mejoras en la velocidad de entrenamiento de la RN

6 CONCLUSIONES

Las propuestas de mejora en la reducción del gasto computacional en el esquema de control estándar presentado han demostrado funcionar satisfactoriamente. Incluso en los casos en que el sistema no lineal a controlar presentase ciertos rangos de funcionamiento inestables, los algoritmos modificados desarrollados para este trabajo han ofrecido buenos resultados, como se ha podido observar en la figura 5.

La integración de ambas Redes Neuronales en dicho esquema de control es posible, manteniendo los requisitos que una aplicación de control en tiempo real puede exigir. Los periodos de entrenamientos se han reducido notablemente gracias al nuevo enfoque presentado en este trabajo.

Como futuro trabajo queda la finalización de las pruebas para realizar un estadístico que demuestre de forma fehaciente las posibilidades de esta metodología presentada. Además se estima de gran interés el validarlo con un sistema real, donde las exigencias temporales sean lo suficientemente estrictas como para ofrecer ventajas frente a otras propuestas.

Referencias (10 ptos, negrita)

- [1] Camacho, E. F., y Bordons, C., (2005) "Nonlinear Model Predictive Control: an Introductory Review", International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Germany.
- [2] Chen, S., Billings, S., y Grant, P., (1990) "Non-linear system identification using neural networks," International Journal of Control, vol. 51, pp. 1191-1214.
- [3] Chen, L., y Narendra, K. S., (2004) "Identification and Control of a Nonlinear Discrete-Time System Based on its Linearization: A Unified Framework", Transactions on Neural Networks, 15(3), pp. 663-673.
- [4] Galván, J.B., (1998) "Tuning of optimal neural controllers", Proc. Int. Conf. on Engineering of Intelligent Systems, Spain, pp. 213-219.
- [5] Galván, J.B., (2003) "Comparison of training algorithms for optimal neural controllers", Proc. 21th IASTED International Multiconference, Austria.
- [6] Irigoyen, E., Galván, J.B., y Pérez-Illarbe, M. J., (2003) "A Neural Controller for Constrained Optimal Control of Nonlinear Systems", IASTED International Conference Artificial Intelligence and Applications, Spain.
- [7] Irigoyen, E., Galván, J.B., y Pérez-Illarbe, M. J., (2003) "A Neuro Predictive Controller for Constrained Nonlinear Systems", IASTED International Conference Artificial Intelligence and Applications, Spain.
- [8] Lera, G., y Pinzolas, M. (2002) "Neighborhood Based Levenberg-Marquardt Algorithm for Neural Network Training", Transactions on Neural Networks, 13(5), pp. 1200-1203.

- [9] Narendra, K. S., y Lewis, F. L., (2001) "Special issue on neural network feedback control", *Automatica*, 37(8).
- [10] Narendra, K. S., y Parthasarathy, K., (1990) "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27.
- [11] Pirabakaran, K, y Becerra, V., (2002) "PID Autotuning using Neural Networks and Model Reference Adaptive Control", 15th Triennial IFAC World Congress, Spain.
- [12] Toledo, A., Pinzolas, M., Ibarrola, J., y Lera, G. (2005) "Improvement of the Neighborhood Based Levenberg-Marquardt Algorithm by Local Adaptation of the Learning Coefficient", *Transactions on Neural Networks*, 16(4), pp. 988-992.

STATE MACHINE MODELLING AND SIMULATION IN PROGRAMMABLE LOGIC DEVICES

Rui Antunes

Departamento de Engenharia Electrotécnica

Escola Superior de Tecnologia de Setúbal do Instituto Politécnico de Setúbal

Rua Vale de Chaves, Estefanilha, 2910-761 - Setúbal, PORTUGAL

Phone: +351 265 790 000; fax: +351 265 721 869; email: rantunes@est.ips.pt

Abstract

This article presents the use of the StateCad application tool to perform digital State Machine design models and simulation for Programmable Logic Devices, such as Xilinx CPLDs and FPGAs. The enormous advantages of this powerful application are described on this article, with several examples. The use of Digital State Machine modelling tools quickly decreases the prototype times, designing faster any State Machine digital circuit, on a single programmable logic device chip.

Key Words: Digital Simulation, State Machines, Programmable Logic Devices.

1 INTRODUCTION

Today the use of programmable logic devices (PLDs) is widely diffused in digital electronic systems. The most used are CPLDs (Complex programmable Logic Devices) and FPGAs (Field Programmable Gate Arrays). These electronic programming devices are becoming widely spread due to its low-cost and its non-volatility. Xilinx [1] and Altera [2] are the PLDs world class major suppliers. Figure 1 shows the main features of Xilinx CPLDs and FPGAs:

The Ultimate System Integration Platform 14K to 330K logic cells	
World's Lowest Cost FPGA 50K to 5 million system gates	
High Volume, Low Power CPLD 32 to 512 Macrocells	
Lowest Cost per Macrocell 36 to 288 Macrocells	

Figure 1: Xilinx CPLDs and FPGAs family

One standard low-cost Xilinx CPLDs family is the In-System Programmable CPLD XC-9500 [3]. This family allows 10.000 erase/programming cycles, has 36 up to 576 macrocells, 800 up to 6400 usable gates, and contains 36 up to 288 registers. It has also IEEE 1149.1 JTAG compatibility, for In-System programming. Table 1 shows its main features:

Table 1: Xilinx XC-9500 family.

CPLD	Characteristics
XC9536	800 usable gates, 36 Registers, 100Mhz Operating frequency.
XC9572	1600 usable gates, 72 registers, 125Mhz Operating frequency.
XC95108	2400 usable gates, 108 registers, 125 Mhz Operating frequency.
XC95144	3200 usable gates, 144 registers, 125 Mhz Operating frequency.
XC95216	4800 usable gates, 216 Registers, 111 Mhz Operating frequency.
XC95288	6400 usable gates, 288 Registers, 92 Mhz Operating frequency.

2 THE STATECAD APPLICATION

Special software applications are used to design, simulate and program a digital circuit in a programmable logic device chip. The application Xilinx ISE is one of the most popular used. It is today, the industry's most complete programmable logic design solution for PLDs optimal performance, cost reduction, power management and productivity. The StatedCad [4] application comes within Xilinx ISE Foundation (Project Navigator) [5], and allows the design and simulation of digital State-Machines. The StatedCad application can design any state machine that models a system. It also allows a state

machine simulation, the VHDL/Verilog automatic code generation, and finally, the PLD programming. StateCAD is a visual representation of a state diagram. The state diagrams visually describe the behaviour of a sequential machine. StateCAD lets us layout a sequential machine, defining outputs, inputs, and transitions between states. When the diagram is complete, the software automatically translates the visual representation into a Hardware Description Language (HDL). For using StateCAD, we will not need to understand how to translate from a state diagram into sequential logic (StateCAD does it all).

2.1 MATHEMATICAL MODEL OF A FINITE STATE MACHINE (FSM)

An acceptor state machine is a quintuple $(\Sigma, S, S_0, \delta, F)$, where:

- Σ is the input, a set of symbols.
- S is a set of states.
- S_0 is the initial state, belonging to S .
- δ is the state transition function: $\delta: S \times \Sigma \rightarrow S$.
- F is the set of final states, belonging to S .

A transducer state machine is a six tuple $(\Sigma, \Gamma, S, S_0, \delta, \omega)$, where:

- Σ is the input, a set of symbols.
- Γ is the output, a set of symbols.
- S is a set of states.
- S_0 is the initial state, belonging to S .
- δ is the state transition function: $\delta: S \times \Sigma \rightarrow S \times \Gamma$.
- ω is the output function.

The Mealy machine exists when the output function is a function of a state and input ($\omega: S \times \Sigma \rightarrow \Gamma$). The Moore machine exists when the output function depends only on a state ($\omega: S \rightarrow \Gamma$). The number of states in a Moore machine will be greater than or equal to the number of states in the corresponding Mealy machine.

In order to optimize an FSM (Finite State Machine) we must obtain the machine with the minimum number of states that equals the same function. This can be achieved using a colouring algorithm.

In a digital circuit, a FSM may be built using a programmable logic controller, a programmable logic device, logic gates and flip flops. A digital hardware implementation requires a register to store state-machine variables, a combinational logic block for obtaining the state transition, and a second block of combinational logic that allows obtaining the output of a FSM [6].

2.1.1 The 4-Module counter model

Let's consider as an example, a four module digital counter, modelled with a State-Machine. Figure 2 shows the circuit diagram for a 4 bit TTL counter, a simple type of a finite state machine:

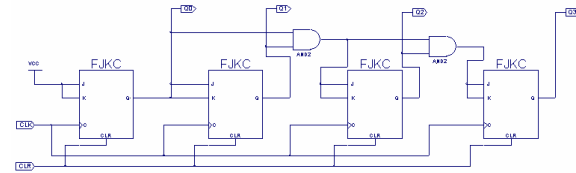


Figure 2: 4 bit TTL counter digital circuit

The State-Machine model for the synchronous 4-module counter will be created with StateCAD, as an application modelling and simulation example. The four ellipses are states, and the arrows the correspondent transitions. Each transition has one condition associated (in this case, the input INP), which sets the next state to be the active. There is also a RESET input, which can reset the State-Machine, making the STATE0 to start as the active state. The Output vector (OUTP) has 2 bits, for generating "00", "01", "10" and "11" possible values for the 4-module counter.

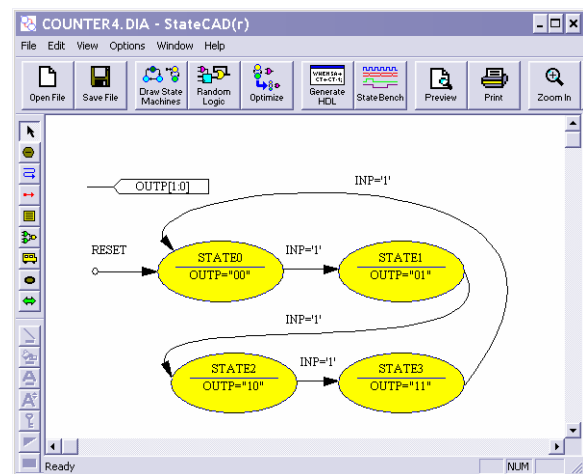


Figure 3: State-Machine diagram of a 4-module digital counter

2.1.2 StateBench simulation

After successfully drawing and compiling the correspondent file (COUNTER4.DIA), with the Option "Generate HDL", we can simulate the behaviour of the 4 State-Machine model, using the StateBench option. Figures 3 and 4 shows the 4-counter working, as clock signal rises.

The StateCAD application allows also a visual simulation of the State-Machine evolution. Figure 4

shows STATE1 as the active (yellow) state, after transition 1 fires. In Figure 6 we can see the final configuration of the State-Machine, stopped at STATE0, after the INP input is reset.

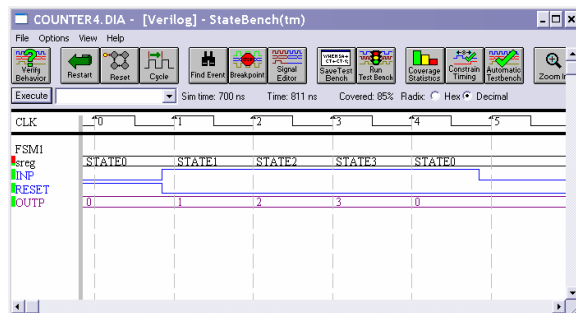


Figure 4: StateBench Simulation of the 4 State-Machine synchronous counter

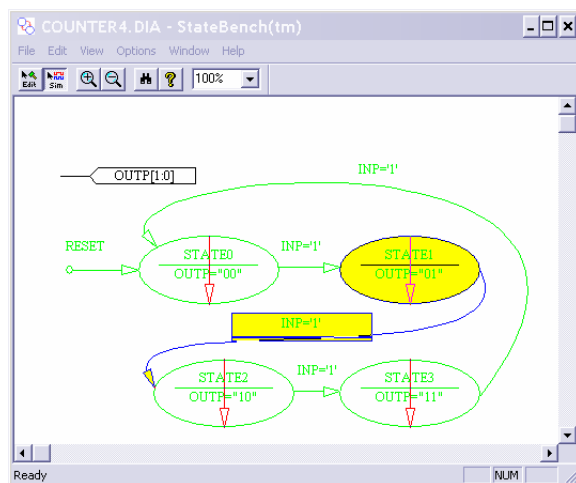


Figure 5: StateCAD visual simulation (STATE1 active)

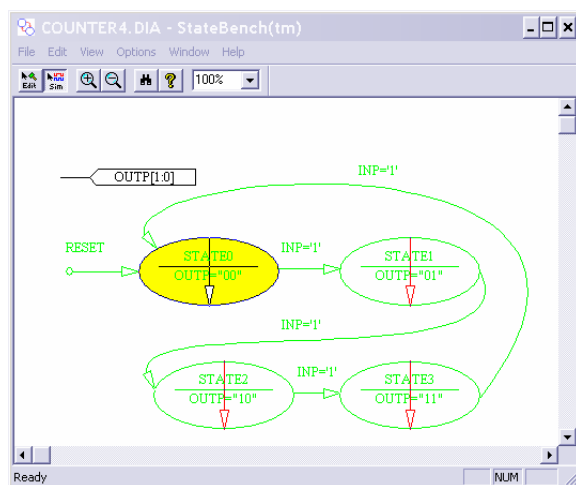


Figure 6: Final State-Machine configuration

2.1.3 HDL and Verilog generation

StateCAD allows also the automatic Verilog [7] code generation of the State-Machine model design (option “Generate HDL”). For the 4-module counter example, the Verilog code generated is:

```
// C:\PROGRAMAS\XILINXISE61\...\COUNTER4.v
// Verilog created by Xilinx's StateCAD 5.03
// This Verilog code (for use with Xilinx XST) was generated
// using:
// binary encoded state assignment with structured code format.
// Minimization is enabled, implied else is enabled,
// and outputs are speed optimized.
```

```
`timescale 1s/1s
```

```
module shell_counter4(CLK,INP,RESET,OUTP0,OUTP1);
```

```
// define the State-Machine inputs and outputs
input CLK;
input INP,RESET;
output OUTP0,OUTP1;
```

```
// register with the output State-Machine data
reg [1:0] OUTP;
reg OUTP0,next_OUTP0,OUTP1,next_OUTP1;
reg [1:0] sreg;
reg [1:0] next_sreg;
```

```
`define STATE0 2'b00
`define STATE1 2'b01
`define STATE2 2'b10
`define STATE3 2'b11
```

```
// This code generate the outputs and the next-state after the clock
// edge.
```

```
always @(posedge CLK or negedge RESET)
begin
  if ( ~RESET ) begin
    sreg=`STATE0;
    OUTP1 = 0;
    OUTP0 = 0;
  end else
  begin
    sreg = next_sreg;
    OUTP1 = next_OUTP1;
    OUTP0 = next_OUTP0;
  end
end
```

```
// This code generate the state of the output variables and the
// next-state, when there are changes in the states, INP, or in the
// OUTP register.
```

```
always @ ( sreg or INP or OUTP )
begin
  next_OUTP0 = 0; next_OUTP1 = 0;
  OUTP=2'h0;

  next_sreg=`STATE0;

  case (sreg)
    `STATE0 : begin
      if ( ~INP ) begin
        next_sreg=`STATE1;
        OUTP= 'h1;
      end
    end else begin
      next_sreg=`STATE0;
      OUTP= 'h0;
    end
  end
end
```



```

        `STATE1 : begin
            if ( ~INP ) begin
                next_sreg=`STATE2;
                OUTP= 'h2;
            end
            else begin
                next_sreg=`STATE1;
                OUTP= 'h1;
            end
        end
        `STATE2 : begin
            if ( ~INP ) begin
                next_sreg=`STATE3;
                OUTP= 'h3;
            end
            else begin
                next_sreg=`STATE2;
                OUTP= 'h2;
            end
        end
        `STATE3 : begin
            if ( ~INP ) begin
                next_sreg=`STATE0;
                OUTP= 'h0;
            end
            else begin
                next_sreg=`STATE3;
                OUTP= 'h3;
            end
        end
    endcase

    next_OUTP1 = OUTP[1];
    next_OUTP0 = OUTP[0];
end
endmodule

// Input module for the State-Machine.
module counter4(OUTP,CLK,INP,RESET);

    output [1:0] OUTP;
    input CLK;
    input INP,RESET;

    wire [1:0] OUTP;
    wire CLK;
    wire INP,RESET;

    shell_counter4
    part1(.CLK(CLK),.INP(INP),.RESET(RESET),.OUTP0(
    OUTP[0]),
        .OUTP1(OUTP[1]));

endmodule

```

Figure 7: Verilog code of the 4-module counter State-Machine model

2.1.4 Schematic symbol creation

We can also create a new symbol (schematic component) of our State-Machine model (option “Create Schematic Symbol”, in Project Navigator (Xilinx ISE Foundation)). The symbol can then be used as an new useful component is ISE Foundation. The next figure shows the symbol (counter4) automatically created from the Verilog code of the 4-module counter State-Machine model, in the ECS

Schematic Editor of Project Navigator (Xilinx ISE Foundation):

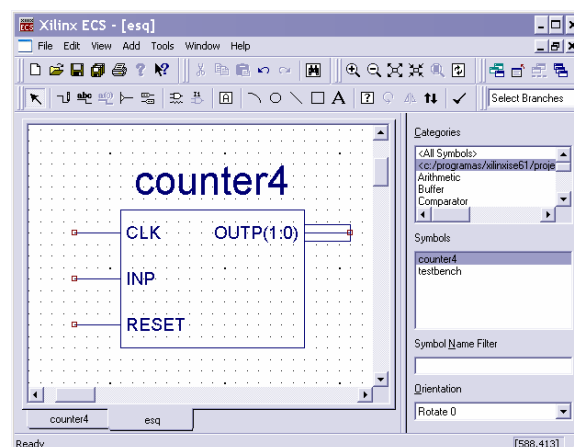


Figure 8: Symbol counter4 automatically created

We can use this new symbol in ECS Schematic Editor, and connect it to any digital schematic, with other pre-defined (library) components, building our final schematic:

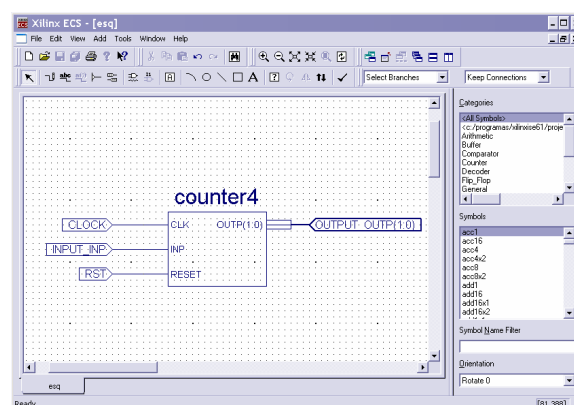


Figure 9: A complete circuit using counter4 symbol

2.1.5 Simulation

Also, we can simulate the whole digital circuit with the ModelSim Xilinx Edition [8], that can be used with the Xilinx ISE Foundation software.

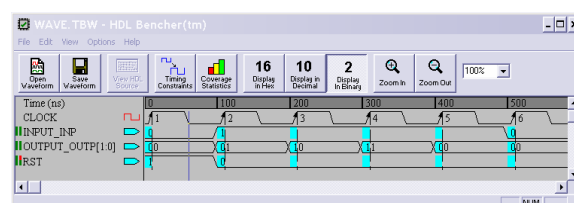


Figure 10: ModelSim simulation of the 4-counter circuit

2.1.6 PLD Pinout definition

The next phase is assigning the PLD pins (CPLD XC-9536) to the inputs and outputs of the 4-module counter model. For that, we use the ChipViewer application (inside Project Navigator). Using a drag&drop method we can assign any I/O CPLD pin to the inputs CLK, INP and RESET, and also to the outputs OUTP<0> and OUTP<1>.

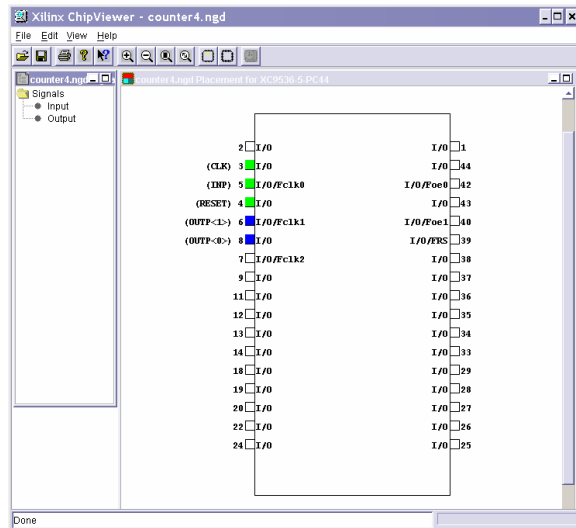


Figure 11: ChipViewer application

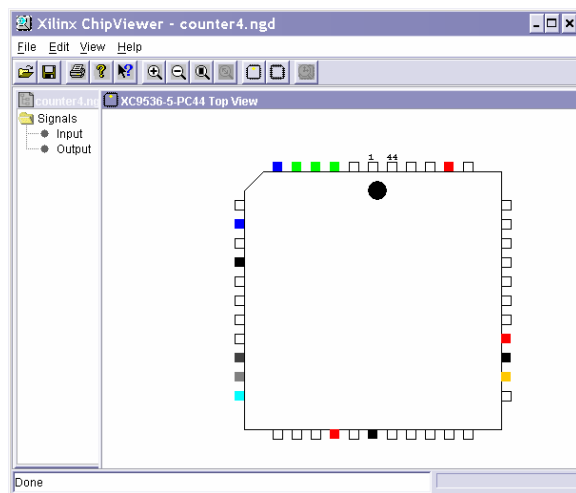


Figure 12: CPLD XC-9536 pinout

2.1.7 PLD Programming (iMPACT)

Finally, we will need to program the XC-9536 CPLD in order to generate and perform the 4-module counter model. It is used the iMPACT application [9] (inside ISE Foundation) and a JTAG parallel programming cable, to connect the CPLD with a personal computer (PC). The JTAG cable connects directly to the PC parallel port.

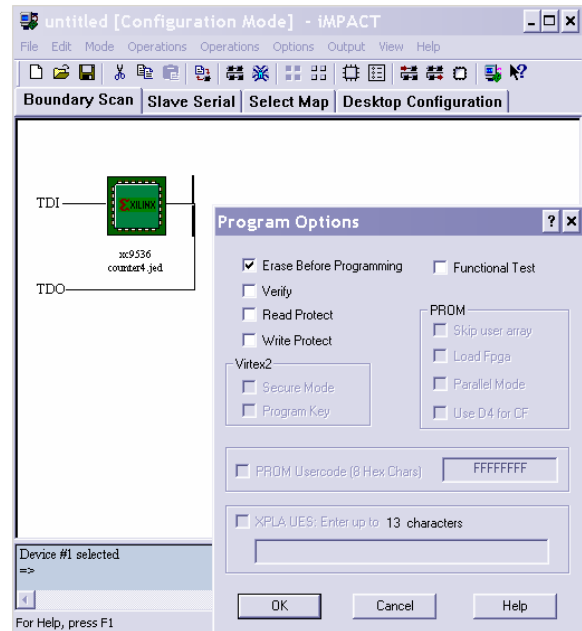


Figure 13: iMPACT application to program the CPLD XC-9536

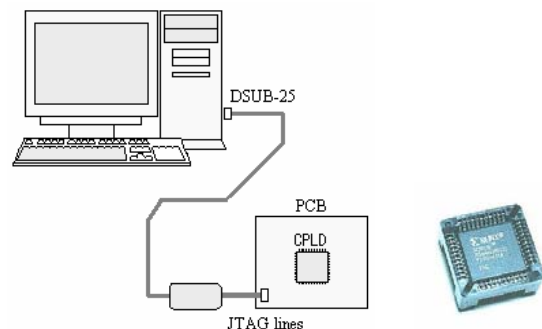


Figure 14: Programming JTAG cable

3 RESULTS

StateCAD lets us layout a sequential machine, defining outputs, inputs, and transitions between states. When the diagram is complete, this software application translates the visual representation into a Hardware Description Language (HDL). It is not necessary to understand the language. By default, StateCAD uses One Hot Encoding. The logic may not be fully minimized because the One Hot Encoding uses more flip flops. The tradeoff, is that the feedback logic is sometimes simpler. For using StateCAD, we will not need to understand how to translate from a state diagram into sequential logic. StateCAD does everything for us, improving the design of Digital State-Machine models into CPLDs and FPGAs prototypes in more than 1000 times faster, when compared with Sequential Finite State-Machine (SMT-FSM) techniques, converting

Mealy and Moore State-Machines to digital sequential logic circuits, with Flip-Flops and the associated logic blocks.

4 CONCLUSIONS

State-Machine modelling and simulation software applications, like StateCAD, can be quite useful to design, simulate and test any Finite State-Machine model, without the need to apply sequential Finite State-Machine techniques, that can be hard to get and very slow to implement in-circuit. These FSM techniques can also bring along human heuristic errors.

The use of the digital programmable logic devices for implementing State-Machine models is becoming the fast and right solution, due to its low-cost, along with its non-volatile capabilities.

Acknowledgements

The author acknowledges the Escola Superior de Tecnologia of Setúbal Polytechnic Institute (Portugal), its Department of Electrical Engineering, and the Digital Systems Labs., for using the ISE Foundation software.

References

- [1] <http://www.xilinx.com>.
- [2] <http://www.altera.com>.
- [3] Xilinx Inc., (2005) “Xilinx CPLDs. The Complete CPLD Solution.”, http://www.xilinx.com/products/silicon_solutions/cplds/xc9500_series/index.htm.
- [4] Oklahoma State University, (2006) “Digital Logic Design StateCAD Tutorial”.
- [5] University of California at Berkeley College of Engineering, (2004) “Using Project Navigator”, EECS 150.
- [6] Taub, H., (1983) “Digital Circuits and Microprocessors”, McGraw-Hill Education.
- [7] Brown, S., Vranesic, Z., (2003) Fundamentals of Digital Logic with Verilog design, McGraw-Hill.
- [8] ModelSim User’s Manual, Model Technology Incorporated, 2005.
- [9] Xilinx ISE In-Depth Tutorial, Xilinx, 2005.

SISTEMA DE CONTROL BASADO EN PC HOLCIM CARBONERAS

Vicente Camacho Molero
Vicente.Camacho@holcim.com

Francisco Javier González Craviotto
FJavier.Gonzalez@holcim.com

1 SISTEMA DE CONTROL DE LA FÁBRICA DE CARBONERAS

1.1 SISTEMA DE CONTROL BASADO EN PC

En una océano industrial automatizado con PLC's, nuestro sistema de control es una isla controlada por Pc's.

En la actualidad no se puede afirmar que nuestro sistema sea mejor ni peor que los que utilizan PLC, simplemente es diferente. En muchos casos se ha comentado que este es el futuro, pero siempre la primera pregunta que nos sugieren cuando comentamos la singularidad de este sistema es ¿no se cuelgan los ordenadores? Pues la verdad, y aunque parezca mentira, es que no. El secreto quizá esté en que en las estaciones de proceso, donde correría la lógica de los PLC, sólo tienen instalado el Windows 2000 y la aplicación Bridgeview, no utilizándose para otras tareas, ni instalándose programas distintos a los ya mencionados.

1.2 SISTEMA A MEDIDA.

El sistema de control con el que está automatizada gran parte de nuestra Fábrica ha sido desarrollado por la empresa española EYPSCAP.

Debido a nuestros requerimientos EYPSCAP realizó un programa a medida. Éste fue el primer trabajo realizado por esta compañía basándose totalmente en PC para implementar un sistema de control.

Al ser un programa piloto, hemos sufrido todo el desarrollo del mismo. Siendo positivos podemos decir que todas las incidencias que hemos tenido en el sistema de control desde sus comienzos nos han permitido conocerlo mejor. Nos ha permitido a la vez detectar deficiencias y subsanarlas con

mejoras que se han desarrollado en el programa a lo largo del tiempo.

Al ser realizado a medida, EYPSCAP tuvo que desarrollar el sistema de acuerdo a los estándares de la compañía, un sistema de control totalmente abierto para el personal de planta, erradicación de elementos intermedios dentro de la instalación (cajas negras), entorno visual según especificaciones.

Este tipo de arquitectura no sólo ha sido implementada en la fábrica de Holcim Carboneras, EYPSCAP también lo ha implementado en otras instalaciones como desaladoras, túneles, etc.

1.2.1 BRIDGEVIEW.

Todo el sistema de control esta desarrollado utilizando el programa BridgeView 2.1. Este sistema fue lanzado por National Instruments para su utilización en entornos industriales en 1999. Realmente es una aplicación basada en Labview para la automatización industrial.

2 ARQUITECTURA DEL SISTEMA.

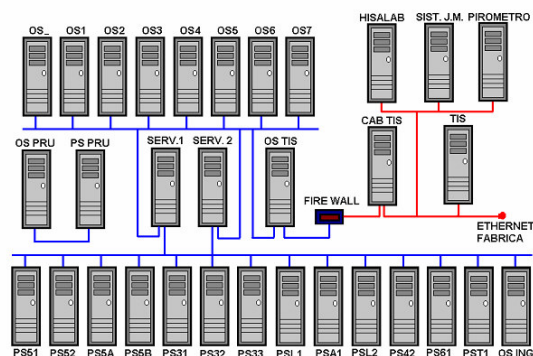


Figura 1: Arquitectura del sistema de control

En la figura 1 podemos observar la arquitectura utilizada en el sistema de control de la fábrica de carboneras.

El sistema se puede dividir en tres grandes grupos de ordenadores, el que forman los servidores, el formado por las estaciones de proceso (PS) y el formado por las estaciones de operador (OS)

2.1 PERIFERIA DISTRIBUIDA

La arquitectura utilizada para el desarrollo de esta aplicación es periferia distribuida, la cual esta formada por grupos de bornas inteligentes instaladas en armarios cerca de los accionamientos a controlar.

En nuestro caso, podemos diferenciar dos grupos fundamentalmente:

Los relacionados con las señales de centro de control de motores, CCM's, (figura 2) en los que las bornas son normalmente a 220 V estando ubicadas al lado de los arrancadores.



Figura 2: Panel de control de CCM con Bornas Wago

Y los relacionados con las señales de campo, (figura 3) en los que las bornas son a 24V estando ubicadas en campo, cerca de los accionamientos.



Figura 2: Cuadro de campo con bornas Wago

La principal ventaja de este tipo de instalaciones es la reducción de costos, pero sin olvidar la mejora sustancial desde el punto de vista del mantenimiento con la eliminación de elementos intermedios.

2.1.1 SERVIDORES

El sistema esta gobernado por dos servidores redundantes. Estos servidores tienen dos tarjetas de red de 1Gb, las cuales están conectadas a sendos Switches también de 1Gb. Disponen de una memoria ram de 1 Gb y de una velocidad de CPU de 3,2 GHz. con lo que se consigue un tiempo de ciclo en torno a los 500 ms.

En cada uno de estos servidores reside la aplicación y ante un fallo o la necesidad de actualizar uno de ellos se puede recurrir al de respaldo sin que el sistema deba pararse y sin sufrir incidencias.

Con el fin de conseguir un sistema más robusto estos dos ordenadores son un poco especiales. Soportan doble disco duro en modo RAID, en forma de espejo, de manera que si hubiera un fallo en uno de ellos el ordenador seguiría funcionando con el otro.

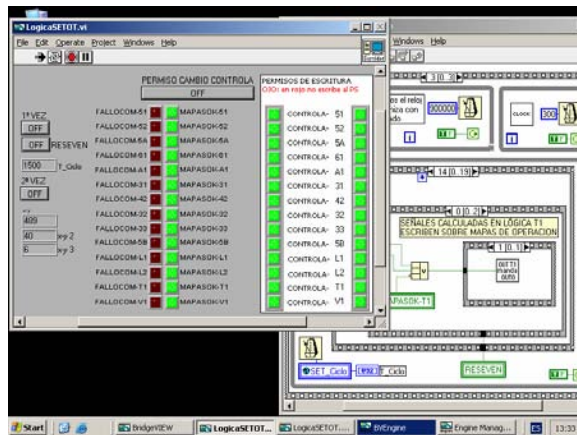


Figura 3: Pantalla de servidor

2.1.2 PS's

En el primer Switch están conectados los servidores y forman la red de PS's. Como hemos dicho anteriormente los PS's vendrían a sustituir al PLC en una configuración ordinaria.

Todos los ordenadores del sistema tienen configuraciones normales y corrientes, como los que tenemos en nuestras casas. Las especificaciones básicas a la hora de la elección de la CPU es que tengan un Slot PCI y al menos 500 Mb de memoria Ram. La media de velocidad de la CPU está actualmente por encima de los 2 GHz, claro está que a mayor velocidad, mejor funcionará la aplicación.

Para mayor comodidad a la hora de gestionar las CPU's de las distintas instalaciones, todos los PS's y servidores son controlados desde un único monitor, ratón y teclado.

Hace unos seis años, al comienzo de nuestra andadura con este tipo de entorno, en el mercado podían encontrarse Pentium III a 500 MHz. Con estos ordenadores se tenía una velocidad de ciclo de aplicación de unos 300ms. Estas CPU's se han ido sustituyendo a lo largo del tiempo, en la actualidad, con los nuevos ordenadores que son reemplazados sistemáticamente cada 3 o 4 años, el tiempo de ciclo de la aplicación está en unos 100 ms.

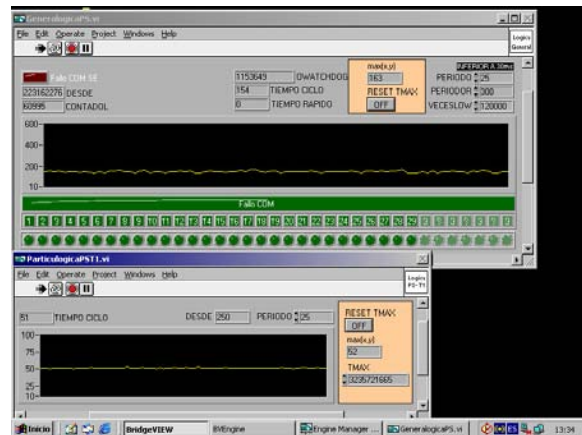


Figura 2: Pantalla de PS

2.1.3 OS's

Los PS's necesitan el slot PCI para poder insertarles una tarjeta de adquisición de datos que se comunica con las señales de campo a través de red profibus. En nuestro caso esta tarjeta es una tarjeta de la marca Applicom del tipo PCI/PCU 1500 PFB, para profibus DP, con un Baud Rate seleccionado de 93750.

Al segundo de los Swiches está conectado a los servidores y forma la red de OS's. Los OS's equivaldrían a las pantallas donde se visualiza el SCADA de una instalación típica basada en PLC.

En la red de OS's podemos encontrar ordenadores similares a los de la red de PS's, la única diferencia con respecto a ellos es la memoria RAM. En los OS's es de 1Gb, aunque la tendencia es a poner los PS's también con 1Gb.

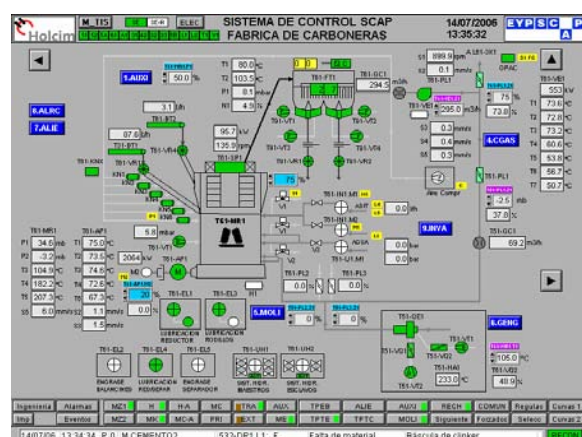


Figura 3: Pantalla de OS

En la figura 3 podemos ver una pantalla típica de los OS's.

Desde un Os se puede acceder a cualquier instalación, teniendo distintos niveles de acceso a las mismas, relativos a arranque de maquinas cambios de parámetros de proceso, parámetros de control.... Estos niveles de acceso son configurables mediante clave.

2.1.4 BUS DE CAMPO

La comunicación entre la tarjeta Applicom situada en los PS's y las bornas inteligentes se realiza mediante fibra óptica y cable. Por especificaciones del proyecto siempre que la distancia hasta las señales de campo supera un cierto valor se utiliza fibra óptica.

Cada PS controla una parte de la instalación. Por lo que hay pequeñas redes colgando de cada tarjeta de comunicaciones. Como la tarjeta de comunicaciones Applicom tiene comunicación profibús, utilizamos convertidores profibús – fibra óptica. Actualmente podemos encontrar instalados convertidores de las marcas SCAP, Westermo y Siemens, siendo estos últimos los que mejores resultados están dando, de momento.



Figura 4: Armario de convertidores

2.1.4.1 BORNAS E/S

Las bornas inteligentes utilizadas para la adquisición de datos en la palnta de Holcim Carboneras son de la marca Wago. Aún teniendo un extenso catálogo,

nosotros hemos preferido estandarizar, utilizando siempre los mismos modelos desde el comienzo de la instalación. Serie 750.

Así pues podemos encontrar las siguientes bornas:

Nombre	Descripción
750-301	Cabecera Wago (A-D)
750-323	Cabecera Wago (D)
750-600	Terminadora
750-400	Entrada Digital a 24 V
750-405	Entrada Digital a 220 V
750-466	Entrada Analógica activa
750-454	Entrada Analógica pasiva
750-554	Salida Analógica
750-513	Salida Digital
750-610	Borna fusible a 24 V
750-611	Borna fusible a 220V
750-669	Termopar

Tabla 1: Bornas más usuales dentro del sistema de control

En la tabla 2 podemos ver las bornas más representativas del sistema de control. Tienen características similares a las anteriores pero están diseñadas para su utilización en zonas con riesgo de explosión. Cumplen con tanto con la normativa ATEX.

Nombre	Descripción
750-600	Terminadora
750-438	Entrada Digital a 24 V
750-485	Entrada Analógica pasiva
750-535	Salida Digital
750-625	Borna fusible a 24 V

Tabla 2: Bornas ATEX

Con la utilización de estas bornas se ha tendido a la eliminación de cajas negras. Eliminando los puntos intermedios entre los sensores y los adquisidores de señal. Todo lo que no está no tiene la opción de fallar.

Se ha reducido el tamaño de los paneles, se ha reducido el número de conexiones y se ha disminuido el costo de reposición de tarjetas, ya que son tarjetas de dos canales.

Por el contrario una desventaja de esta borna es que a la hora de sustituirlas deja sin comunicación la barra en la que está insertada, pudiendo en la mayoría de los casos provocar un paro de la instalación de la que forma parte

3 PROBLEMAS CON LA UTILIZACION DE PC's

El mayor problema que se ha tenido a la hora de la automatización basada en PC ha sido la inexperiencia. La utilización de un lenguaje de programación no usado anteriormente para controlar este tipo de instalaciones.

El aumento del número de instalaciones a controlar ha sido también un inconveniente. De un sistema pensado inicialmente para controlarse con unos 10 ordenadores, se dispone al día de hoy de más del doble, y aún queda buena parte de la fábrica por automatizar.

Hemos de reseñar que la instalación actual tiene más de 11.000 señales manejadas por el sistema de control, con unos 50 lazos de control. De las 11.000 señales del sistema de control, 5.018 son entradas digitales a 24 V, 2.728 son entradas digitales a 220 V, 1.076 son entradas analógicas, 2028 son salidas digitales y 148 son salidas analógicas. Con estas señales de controlan mas de 900 accionamientos.

La red de PS está dividida en 14 instalaciones cada una de las cuales controla una sección de fábrica. La mayor de nuestras subredes (la perteneciente al molino de escoria) direcciona unos 46 nodos profibús, de los cuales 28 son cabeceras wago, siendo el resto basculas pesadoras marca Shenck y dos PLC's Siemens 312 2DP, funcionando como subsistemas.

3.1 MANTENIMIENTO

Uno de los objetivos a la hora de automatizar una instalación es la estandarización. Por ello todos los arrancadores han sido agrupados en grandes grupos según sus características, como pueden ser motores reversibles, compuertas neumáticas, etc.

Desde el punto de vista de mantenimiento se intenta conseguir que cualquier problema detectado en una instalación quede definido por un aviso o alarma registrado en el sistema de control, reduciendo al máximo el tiempo de detección de averías y por tanto de parada. Así pues, como fallos o alarmas generales en una máquina podemos encontrarlos; fallo de protecciones, fallo de respuesta de marcha, paro local sin permiso, contactor bloqueado, fallo de control de rotación, fallo de variador, desvío, tirón, nivel, fallo de final de

careas, etc. También existen alarmas generadas en función de las señales analógicas que se registran.

3.2 SISTEMA AISLADO

Unos de las principales amenazas a las que está expuesto el sistema de control es a la infección por virus informáticos.



Figura 5: Armarios PS_OS_SE

Es éste el principal motivo por el que se intenta aislar en la medida de lo posible contra ataques externos mediante cortafuegos. El sistema a su vez está totalmente desligado de la red informática general de fábrica.

Es sabido que la temperatura también afecta negativamente a las CPU's. Todas las que conforman el sistema de control se encuentran en la Sala de Ingeniería, una sala climatizada con aire acondicionado y limpia junto a la Sala de Control de fábrica. Los operadores que manejan las instalaciones desde la Sala de Control sólo disponen de teclado monitor y ratón, los cuales están conectados remotamente mediante dispositivos interlink.

IDENTIFICACIÓN BAYESIANA ORIENTADA AL CONTROL ROBUSTO

Rosa M^a Fernández-Cantí

Grup de Teoria de Control, Dpt. Teoria del Senyal i Comunicacions, Univ. Politècnica de Catalunya (UPC)

Jaume Herranz Luis

Grup de Teoria de Control, Dpt. Teoria del Senyal i Comunicacions, Univ. Politècnica de Catalunya (UPC)

Resumen

Este trabajo considera la obtención de modelos para la síntesis de controladores robustos. Se presenta una metodología bayesiana que combina aspectos de las actuales técnicas de identificación robusta, tanto deterministas como estocásticas. El enfoque bayesiano presenta interesantes ventajas tanto a nivel conceptual y metodológico como de resultados. Se comentan aspectos de su implementación por medio de técnicas de simulación de Monte Carlo/cadenas de Markov y se ilustra parte de la metodología con un ejemplo.

Palabras Clave: Robust control oriented identification, Bayesian decision theory, Markov chain Monte Carlo (MCMC)

Las técnicas de identificación robusta van más allá de las tradicionales técnicas de predicción de error (*prediction error methods*) [8]. Éstas fueron descartadas desde un primer momento por ser aplicables sólo al caso de incertidumbre en los parámetros (y no tener en cuenta la incertidumbre en la propia estructura del modelo) y por dar lugar a cotas probabilísticas en principio incompatibles con la filosofía del control robusto. Este segundo argumento ha perdido fuerza recientemente con la aparición de los llamados controladores de riesgo ajustado (*risk adjusted*), esto es, controladores probabilísticamente robustos [16]. Por otro lado, nunca hubo consenso absoluto al respecto de la interpretación de las regiones de incertidumbre deterministas también llamadas de peor caso (*worst case*).

En términos generales, las actuales técnicas de identificación robusta pueden clasificarse en deterministas y estocásticas. Entre las primeras destacan las conocidas con el término *set membership* (SM) de las cuales destacan, a su vez, los algoritmos interpolatorios, llamados así porque utilizan técnicas de interpolación tales como la de Nevanlinna-Pick (para datos en el dominio frecuencial) y la de Carathéodory-Féjer (para datos en el dominio temporal) [2], [10], [15]. Entre las segundas, destacan las extensiones de [8] para la obtención de modelos de error (*model error modelling*) y la técnica conocida como *non-stationary stochastic embedding* (NSSE) donde el modelo de error se implementa por medio de un proceso de Wiener cuya varianza aumenta con la frecuencia [4], [16].

La investigación actual en el campo de la identificación robusta se centra en la obtención de cotas de incertidumbre más ajustadas y en el desarrollo de algoritmos más eficientes para su cálculo. En general, las técnicas deterministas son computacionalmente intensivas pero permiten la entrada explícita en el proceso de identificación de información *a priori* sobre la planta y sobre el ruido de medida. Ello se implementa por medio de la

1 INTRODUCCIÓN

En la actualidad, los métodos de síntesis de controladores robustos han alcanzado un elevado grado de madurez. En particular, la síntesis de controladores \mathcal{H}_∞ y μ está ya muy sistematizada y la carga del problema ha pasado de la fase de solución a la de formulación, es decir, a la selección de modelos adecuados para la planta a controlar y a la determinación de las especificaciones de diseño [15]. El control robusto presenta la particularidad de que no basta con proporcionar un modelo nominal de la planta sino que también hay que proporcionar un modelo de la incertidumbre del modelo nominal. El objetivo de la identificación de sistemas orientada al control robusto, también llamada identificación robusta, es obtener ambos modelos, nominal y de incertidumbre. Cuantificar bien la incertidumbre del modelo es sumamente importante. Una modelización inadecuada puede dar lugar a controladores frágiles, como muestra el interesante ejemplo en [12]. Mucho más comúnmente, una cuantificación pesimista de la incertidumbre da lugar a diseños excesivamente conservadores que penalizan el comportamiento del sistema en aras de conseguir un nivel de robustez que en realidad no es necesario.

definición del conjunto de consistencia (*consistency set*), también llamado conjunto de modelos factibles (*feasible model set*). Por el contrario, las técnicas estocásticas no necesitan tantos recursos de computación y dan lugar a bandas de incertidumbre más pequeñas; no obstante, a veces pueden presentar problemas de sesgo dado que la posible información *a priori* de la planta no se utiliza eficientemente.

En esta comunicación se presenta una metodología estocástica que permite la introducción de la información *a priori* de la planta y del ruido de medida en el proceso de modelización. Ello se consigue por medio de la definición de un conjunto bayesiano de modelos creíbles \mathcal{B} inspirado en el conjunto de consistencia de los métodos deterministas. De esta manera la metodología retiene las propiedades más interesantes de los dos enfoques, determinista y estocástico. Para el desarrollo completo, así como para una recopilación de las técnicas de identificación robusta mencionadas, véase [3]. En cuanto a la implementación de los conceptos bayesianos, esto ha sido posible sólo recientemente gracias a la aparición y desarrollo en la última década de las técnicas conocidas como MCMC (Markov chain Monte Carlo) [5]. Para una panorámica de la filosofía y herramientas bayesianas véase, por ejemplo [1], [7], [14]; y, para otros trabajos relacionados con la identificación bayesiana orientada a control, véase [6], [9], [13].

2 PRELIMINARES

2.1 EL PROBLEMA

En el caso de incertidumbre aditiva el conjunto de modelos \mathcal{M} que contiene el modelo nominal y todas sus posibles variaciones (*perturbations*) compatibles con la información disponible sobre la planta real puede expresarse como

$$\mathcal{M} = \{G_0 + \Delta \cdot W_a : |\Delta(\omega)| \leq 1, \forall \omega\}, \quad (1)$$

donde G_0 es la función de transferencia del modelo nominal, W_a es la función de transferencia del error de identificación y Δ es una función de transferencia cualquiera que satisfice la condición $|\Delta(\omega)| \leq 1$. Esta condición permite describir la incertidumbre de la respuesta frecuencial del modelo en forma de círculos centrados en el valor nominal (*disk uncertainty*). ΔW_a es el llamado modelo de error.

El problema de la identificación robusta consiste en obtener el modelo nominal G_0 y la cota de error W_a a partir de datos experimentales de entrada/salida $\{u_n, y_n\}_{n=0}^{N-1}$ y de posible información *a priori* sobre la

planta “real” G_{real} y el ruido de medida $\{v_n\}_{n=0}^{N-1}$ [2], [15]. En el caso de las técnicas deterministas esta información *a priori* define la clase de modelos \mathcal{G} (*model class*) y la clase de perturbaciones \mathcal{V} (*disturbance class*). Las formulaciones más típicas incluyen descripciones *unknown-but-bounded* (UBB) del tipo $|v_n| \leq \delta, \forall n$, para el ruido de medida y $h_n \leq L\rho^n, \forall n, L > 0, 0 \leq \rho \leq 1$, para la respuesta impulsional de la planta.

No sólo interesa que el conjunto de modelos sea consistente con los datos experimentales y la información *a priori* (lo que supone tener que validar el conjunto de modelos, es decir, verificar su consistencia), sino que, además, interesa que las funciones de transferencia sean de orden reducido (para evitar controladores de orden elevado) y que las cotas de incertidumbre estén bien ajustadas a la realidad y no sean excesivamente pesimistas.

2.1 DATOS EXPERIMENTALES

En el caso de sistemas SISO (*single input single output*) lineales e invariantes en el tiempo, los datos experimentales de entrada/salida en el dominio temporal están relacionados según

$$y_n = G(q, \theta)u_n + v_n, \quad n = 0, 1, 2, \dots, N-1, \quad (2)$$

donde q^{-1} es el operador retardo y $G(q, \theta)$ es una función de transferencia que depende del vector de parámetros θ de dimensión d . También es posible trabajar con datos experimentales de entrada/salida en el dominio frecuencial. En ese caso la señal de entrada es periódica y la salida la constituye la estimación de la respuesta frecuencial de la planta, $\hat{G}(\omega)$, para cada frecuencia de la excitación, $\hat{G}(\omega_m) = G(\omega_m, \theta) + w_m, m = 0, 1, 2, \dots, M-1$.

Como es habitual en la identificación robusta, consideraremos que el modelo de la planta admite la parametrización en función de una serie de funciones de transferencia $B_i(q)$ con denominador fijo.

$$G(q, \theta) = \sum_{i=0}^{d-1} \theta_i B_i(q). \quad (3)$$

De esta manera se evita la estimación de los polos de la planta, muy sensible al ruido de medida. En su lugar, el denominador del modelo se fija *a priori* por medio de $B_i = 1/(q - \xi_i)$ eligiendo los polos $\{\xi_i\}_{i=0}^{d-1}$ en las inmediaciones de los modos dominantes de la planta. Una opción muy extendida consiste en usar funciones B_i tales que formen una base ortonormal en

el espacio de modelos (p. ej. en el espacio \mathcal{H}_2) Esta opción presenta diversas ventajas, véase [11], y puede implementarse mediante la base ortonormal generalizada definida por las funciones

$$B_i(q) = \left(\frac{\sqrt{1-|\xi_i|^2}}{q-\xi_i} \right) \prod_{k=0}^{i-1} \left(\frac{1-\overline{\xi_k}q}{q-\xi_k} \right). \quad (4)$$

Si los polos ξ_i son reales, las funciones B_i corresponden al desarrollo en serie de Laguerre y, si son complejos conjugados, al desarrollo en serie de Kautz. Si son cero, las funciones resultantes $B_i = q^{-i}$ corresponden a un modelo FIR (*finite impulse response*).

La parametrización (3) permite expresar (2) en forma de regresión lineal

$$y_n = \boldsymbol{\varphi}_n^T \boldsymbol{\theta} + v_n, \quad n = 0, 1, 2, \dots, N-1 \quad (5)$$

donde $\boldsymbol{\varphi}_n^T = (\mathcal{B}_0(q)u_n \dots \mathcal{B}_{d-1}(q)u_n)$. En formato matricial, $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\theta} + \mathbf{v}$, donde $\mathbf{y}^T = (y_0 \dots y_{N-1})$, $\mathbf{\Phi}^T = (\boldsymbol{\varphi}_0^T \dots \boldsymbol{\varphi}_{N-1}^T)$, etc. En el caso frecuencial, $\hat{\mathbf{G}} = \mathbf{\Gamma}\boldsymbol{\theta} + \mathbf{w}$, donde $\hat{\mathbf{G}}^T = (\hat{G}_0^R \hat{G}_0^I \dots \hat{G}_{M-1}^I)$, $\hat{G}_0^R = \text{Re}[\hat{G}(\omega_0)]$, $\mathbf{\Gamma}^T = (\mathbf{B}_0^R \mathbf{B}_0^I \dots \mathbf{B}_{M-1}^I)$, y $\mathbf{B}_m = (\mathcal{B}_0(\omega_m) \dots \mathcal{B}_{d-1}(\omega_m))$, $m = 0, 1, 2, \dots, M-1$.

3 REGIONES BAYESIANAS DE INCERTIDUMBRE

3.1 CONJUNTO BAYESIANO DE MODELOS CREIBLES

Definimos el conjunto bayesiano de modelos creíbles como

$$\mathcal{B} \equiv \{G \in \mathcal{S} : p(G|\mathbf{y}) \geq c(\alpha)\} \quad (6)$$

\mathcal{B} contiene todos los modelos G pertenecientes al espacio \mathcal{S} cuya función densidad de probabilidad *a posteriori* condicionada a los datos experimentales \mathbf{y} , $p(G|\mathbf{y})$, está por encima del umbral $c(\alpha)$ donde $100(1-\alpha)\%$ es el nivel de credibilidad deseado. En el caso del control robusto el espacio \mathcal{S} de interés suele ser \mathcal{H}_∞ o \mathcal{H}_1 . El conjunto \mathcal{B} define regiones de densidad de probabilidad *a posteriori* más alta (*highest posterior density*, HDP), más fáciles de obtener que las regiones de confianza clásicas. El término “creíble” (*credible*) se emplea en la teoría bayesiana a fin de diferenciarlo del concepto clásico de “confianza” (*confidence*) puesto que la

interpretación es diferente [14]. La definición de \mathcal{B} para el caso de datos en el dominio frecuencial es análoga a (6).

3.1.1 Construcción

Construir \mathcal{B} consiste en seleccionar el umbral c y en obtener la función de densidad de probabilidad *a posteriori* de los modelos $p(G|\mathbf{y})$ combinando la información *a priori* y los datos experimentales. El umbral c se escoge de manera que la probabilidad de no incluir la planta real dentro del conjunto creíble sea pequeña.

En cuanto a la información *a priori*, al igual que en los métodos deterministas, es necesario definir una clase de modelos \mathcal{G} y una clase de perturbaciones \mathcal{V} . En nuestro caso, estas clases están caracterizadas por las funciones densidad de probabilidad del modelo, $p(G)$, y del ruido de medida, $p_v(v)$. En un entorno típicamente bayesiano estas funciones son *subjetivas*, y describen cuánto confía el investigador en su información *a priori* sobre el sistema. Un alto grado de ignorancia se expresa por medio de funciones densidad de probabilidad *no informativas*, es decir, planas comparadas con la distribución de probabilidad de los resultados experimentales.

Por su lado, los datos experimentales se introducen en el conjunto creíble \mathcal{B} por medio de la función de verosimilitud (*likelihood function*) de las observaciones \mathbf{y} condicionadas al modelo G , $l(G|\mathbf{y})$. Para sistemas lineales del tipo $y = Gu + v$, la función de verosimilitud presenta la misma forma que la distribución del ruido puesto que $y - Gu = v$. Por tanto, $l(G|\mathbf{y}) \equiv p_v(\mathbf{y}|G)$.

Finalmente, para obtener $p(G|\mathbf{y})$ se aplica la regla de Bayes [1], [14],

$$p(G|\mathbf{y}) = \frac{p_v(\mathbf{y}|G)p(G)}{p(\mathbf{y})} \propto p_v(\mathbf{y}|G) \cdot p(G) \quad (7)$$

donde el factor $p(\mathbf{y}) = \int p_v(\mathbf{y}|G)p(G)dG$ es una mera constante de normalización. Nótese que la fórmula de Bayes describe cómo nuestra opinión previa sobre la planta, $p(G)$, se modifica al tener en cuenta los datos experimentales \mathbf{y} , y se convierte en $p(G|\mathbf{y})$.

Nótese también que es posible aplicar (7) de forma iterativa, es decir, usar la $p(G|\mathbf{y})$ obtenida en un primer experimento como información *a priori* para un segundo experimento. Ello permite obtener de

cotas de incertidumbre más pequeñas a medida que se realizan más experimentos [3].

3.1.2 Casos particulares

Hasta ahora hemos supuesto un espacio de modelos genérico sobre el que definir $p(G)$ y $p(G|y)$. Diferentes descripciones de \mathcal{G} dan lugar a diferentes casos particulares de \mathcal{B} . En particular, si definimos la clase de modelos en base al espacio de parámetros Θ , el conjunto bayesiano de parámetros creíble es

$$\mathcal{B}_\theta \equiv \{\theta \in \Theta : p(\theta | y) \geq c(\alpha)\}. \quad (8)$$

Y, si expresamos \mathcal{B} en términos de la respuesta frecuencial del modelo, podemos definir para cada frecuencia ω_i , $i = 0, \dots, M-1$, una región bayesiana de respuesta frecuencial creíble

$$\mathcal{B}_\omega \equiv \{G(\omega_i) \in \mathbb{C} : p(G(\omega_i) | y) \geq c(\alpha)\}. \quad (9)$$

El conjunto (8) es adecuado para cuantificar la incertidumbre en los parámetros de manera análoga a como se hace en los métodos de predicción de error [8]. Pero también puede generar regiones de peor caso, es decir, regiones que contengan el 100% de los valores posibles para θ , y todos con el mismo nivel de probabilidad. Ello se consigue por medio del uso de distribuciones uniformes. Véase la Figura 1.

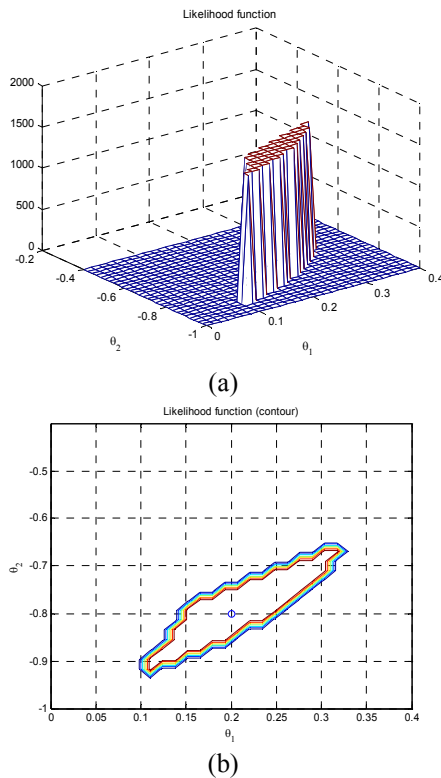


Figura 1: Interpretación de \mathcal{B}_θ como conjunto factible (determinista) de parámetros.

Por su parte, el conjunto (9) es especialmente interesante para el control robusto puesto que las bandas de incertidumbre se obtienen directamente en el dominio frecuencial y, además, al tratarse de distribuciones bidimensionales en el plano de Nyquist su interpretación resulta muy intuitiva. Ver el ejemplo de la sección 4.

3.2 MODELOS JERÁRQUICOS

El uso de modelos de probabilidad jerarquizados (*hierarchical models*) y combinados (*mixture models*) permite obtener modelos complejos a partir de otros más sencillos [14] y permite una gran flexibilidad a la hora de expresar los diferentes aspectos inciertos del modelo. En general, no sólo existe incertidumbre al respecto de los parámetros del modelo sino también al respecto de la propia estructura del modelo. El enfoque bayesiano permite definir conjuntos de modelos tales que contengan estructuras diversas (incluyendo modelos no lineales y de dimensión infinita). Asignando una probabilidad *a priori* a cada estructura en competición y sintonizándola con los datos experimentales por medio de la fórmula de Bayes, es posible obtener las bandas de incertidumbre bayesianas calculando las regiones de probabilidad *a posteriori* más alta. De ahí el interés para el campo de la identificación robusta donde el error de modelización es en gran parte debido a procesos de reducción y simplificación de modelos más complejos. Incluyendo en un mismo conjunto de modelos el modelo nominal (simple) y otros modelos más completos de la dinámica de la planta (no lineales, orden elevado) es posible obtener bandas de incertidumbre alrededor del modelo nominal debidas al resto de modelos más complejos y, por tanto, debidas a la dinámica no incluida en el modelo simplificado.

Para obtener la probabilidad *a posteriori* de un modelo m perteneciente al conjunto de modelos \mathcal{M} , el procedimiento es el siguiente. Cada modelo tiene su propio conjunto de parámetros θ_m perteneciente a la región Θ_m . La distribución *a priori* conjunta de (m, θ_m) se puede factorizar como

$$p(m, \theta_m) = p(\theta_m | m)p(m) \quad (10)$$

donde $p(m)$ es la distribución *a priori* del modelo m . Una vez obtenidos los datos experimentales y , La distribución conjunta *a posteriori* es

$$p(m, \theta_m | y) = \frac{p(y | m, \theta_m) \cdot p(m, \theta_m)}{\int_{\Theta_m} p(y | m, \theta_m) \cdot p(m, \theta_m) \cdot d\theta_m} \quad (11)$$

Sustituyendo (10) en el denominador de (11) tenemos

$$p(m, \theta_m | y) = \frac{p(y | m, \theta_m) \cdot p(m, \theta_m)}{\sum_m \left[\int_{\Theta_m} p(y | m, \theta_m) \cdot p(\theta_m | m) \cdot d\theta_m \right] \cdot p(m)} \quad (12)$$

La función densidad de probabilidad *a posteriori* del modelo m es, finalmente,

$$p(m | y) = \frac{p(m) \cdot \int_{\Theta_m} p(y | m, \theta_m) \cdot p(\theta_m | m) \cdot d\theta_m}{\sum_k p(k) \cdot \left[\int_{\Theta_k} p(y | k, \theta_k) \cdot p(\theta_k | k) \cdot d\theta_k \right]} \quad (13)$$

3.4 IMPLEMENTACIÓN VIA MCMC

La implementación de (13) por métodos numéricos estándar es inviable en el caso de distribuciones de probabilidad arbitrarias y conjuntos de modelos y parámetros moderadamente grandes. Nótese que para calcular $p(m | y)$ hay que obtener la distribución marginal $\int_{\Theta_m} p(y | m, \theta_m) \cdot p(\theta_m | m) \cdot d\theta_m$ para cada $m \in \mathcal{M}$. Por ello hay que recurrir a técnicas de integración de Monte Carlo, en las que se resuelven integrales de la forma

$$I = \int f(\theta) \pi(\theta) d\theta \quad (14)$$

donde $\pi(\theta)$ es una función no negativa, $\pi(\theta) \geq 0$, tal que su integral satisface $\int \pi(\theta) d\theta = 1$. De ahí la interpretación natural de π como función densidad de probabilidad. Los métodos de Monte Carlo obtienen un número N , $N \gg$, de muestras $\{\theta_i\}_{i=1}^N$ distribuidas según $\pi(\theta)$ y aproximan la integral (14) por

$$f_N = \frac{1}{N} \sum_{i=1}^N f(\theta_i) \quad (15)$$

Una manera de generar las muestras de la distribución deseada es hacer que ésta sea la distribución límite de una cadena de Markov. Los métodos conocidos como Markov chain Monte Carlo (MCMC) generan cadenas de Markov cuyos histogramas coinciden con las distribuciones marginales deseadas para cada uno de los parámetros. Es el caso de algoritmos como el muestreador de Gibbs y el algoritmo de Metropolis-Hastings [14].

En el caso de la identificación robusta el problema es que muchas veces desconocemos el número de

parámetros a muestrear (el orden del sistema es incierto) y por ello hay que recurrir a distribuciones de probabilidad combinadas (*mixture distributions*) de la forma

$$\sum_{i=1}^k p_{ik} \cdot p(M_k | \theta_{ik}) \quad (16)$$

donde $p(\cdot | \theta)$ es una densidad parametrizada, la suma de los pesos p_{ik} vale 1 y el número de componentes k es desconocido. En este tipo de problemas hay que usar técnicas del tipo MCMC con salto reversible [5], que permiten muestrear entre espacios de diferente dimensión.

3.5 TEORIA DE LA DECISION

En la sección anterior se ha presentado cómo obtener las bandas de incertidumbre, lo que constituye el punto central de la identificación robusta. Aún así, el punto de vista bayesiano es altamente unificador y permite tratar otros aspectos del problema como son la identificación del modelo nominal, la validación de modelos y el diseño óptimo de experimentos considerándolos como problemas de decisión (para más detalles, ver [3]). En relación a la identificación del modelo nominal puede utilizarse un criterio de máximo *a posteriori* (MAP) pero es mucho más interesante escoger modelos de mínimo riesgo (MR)

$$G_{nom} = \arg \min_G \int_{\mathcal{G}} L(G) p(G | y) dG \quad (17)$$

donde la función de penalización L puede definirse en función de los teoremas de estabilidad robusta y comportamiento robusto [15], reforzando así el carácter de orientación al control robusto; o bien llevar a cabo algún tipo de optimización minimax,

$$G_{nom} = \arg \min_{G(y)} \max_G L(G(y)) \quad (18)$$

Con respecto a la validación de modelos, ésta puede implementarse por medio de un test de hipótesis. El caso bayesiano es de nuevo diferente al caso clásico y un punto central del procedimiento es la definición de los factores de Bayes y la ventana de Occam. Finalmente, con respecto al diseño óptimo de experimentos orientado a identificación robusta, es posible definir una función de utilidad U y obtener el experimento η tal que la maximice. Si se desea que el experimento sea informativo, una opción razonable es maximizar la distancia de Kullback-Leibler entre las distribuciones *a posteriori* y *a priori*,

$$\int \ln \frac{p(G | y, \eta)}{p(G)} p(y, G | \eta) dG dy \quad (19)$$

4 EJEMPLO

Vamos a ilustrar el proceso de obtención de bandas de incertidumbre bayesianas en la respuesta frecuencial de un sistema por medio del uso de un conjunto bayesiano de modelos creíbles (6) formado por diversas estructuras en competición. El caso más simple es suponer que la planta puede modelizarse por medio de un modelo FIR de orden y parámetros inciertos [6].

En primer lugar, e independientemente de los datos experimentales que obtendremos más adelante, realizamos una serie de suposiciones (más o menos razonables) sobre el ruido de medida y sobre la planta desconocida. En cuanto al ruido de medida v suponemos que presenta una distribución gaussiana de media cero y matriz de covarianzas $\sigma_v^2 \times \mathbf{I}$. En cuanto a la planta, “apostamos” *a priori* a que el valor del orden d del modelo está entre 2 y 5, ambos inclusive. Ello define un conjunto de modelos numerable formado por cuatro estructuras: $M_1 \equiv \text{FIR}_2$, $M_2 \equiv \text{FIR}_3$, $M_3 \equiv \text{FIR}_4$ y $M_4 \equiv \text{FIR}_5$. En principio, no hay razón para favorecer la selección de un modelo frente al resto, por lo que a todos les asignamos la misma probabilidad *a priori* de ser seleccionados ($1/4$). Para cada modelo suponemos que la distribución de probabilidad conjunta de sus coeficientes, $p(\theta)$, es normal de valor medio $\theta_0^{(d)} = 0.2 \times \mathbf{1}_d$, $\forall d$. Nótese la jerarquía en la asignación de probabilidades: en el primer nivel seleccionamos la estructura del modelo mientras que en el segundo nivel seleccionamos los parámetros.

El grado de confianza en nuestra “apuesta” de parámetros lo indicamos por medio de la matriz de precisión que, en este ejemplo, se escoge como $\mathbf{R}_0^{(d)} = 100 \times \mathbf{I}_{d \times d}$, $\forall d$. El valor escogido da lugar a una distribución de compromiso, ni excesivamente informativa (picuda) ni excesivamente poco informativa (plana). Al ser $p(\theta)$ normal, las regiones resultantes de incertidumbre en el espacio de parámetros de cada M_i son elipsoides, $(\theta - \theta_0^{(d)})^T \mathbf{R}_0^{(d)} (\theta - \theta_0^{(d)}) \sim \chi_d^2$, donde χ_d^2 es la distribución ji cuadrada con d grados de libertad [8]. Estas regiones pueden traducirse directamente a cotas probabilísticas en el dominio frecuencial. La Figura 2 muestra la función densidad de probabilidad *a priori* $p(M_i, \omega)$ de la respuesta frecuencial de cada uno de los cuatro modelos, M_i , $i=1, \dots, 4$, a la frecuencia $\omega = 1.16 \text{ rad/s}$. Estas cuatro funciones se combinan según

$$p(\omega) = \sum_{i=1}^4 \frac{1}{4} \cdot p(M_i, \omega) \quad (20)$$

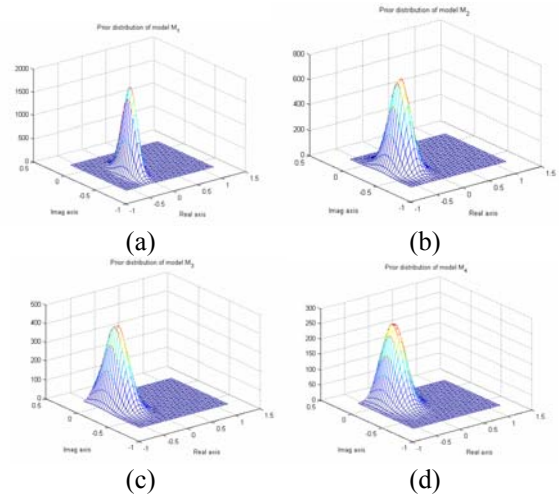


Figura 2: Distribuciones *a priori* $p(M_i, \omega)$, $i=1, \dots, 4$.

a fin de obtener la función densidad de probabilidad *a priori* total $p(\omega)$, para todos los modelos que forman el conjunto creíble (véase la Figura 3(a)). La Figura 3(b) muestra la respuesta frecuencial “real” de la planta desconocida así como el diagrama de contorno de $p(\omega)$ a frecuencia $\omega = 1.16 \text{ rad/s}$. La región correspondiente a una probabilidad creíble del 80% se muestra sombreada. Nótese que todavía no hemos realizado ningún experimento y aún así ya disponemos de unas cotas de incertidumbre en la respuesta frecuencial de la planta desconocida.

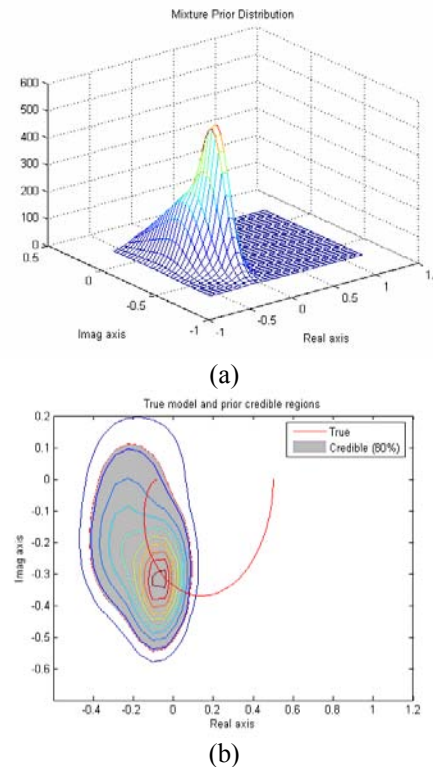


Figura 3: Distribución *a priori* total, $p(\omega)$

Consideremos ahora que la planta supuestamente desconocida es

$$G(s) = \frac{1}{(s+1)(s+2)}. \quad (21)$$

Esta planta se discretiza anteponiéndole un mantenedor de orden cero (*zero-order hold*, ZOH) con periodo de muestreo $T_s = 1s$. A continuación se excita con una señal binaria pseudo aleatoria y se recogen $N = 1000$ muestras de entrada/salida. El ruido de medida es blanco gaussiano de media cero e intensidad 0.01.

Con ayuda de la regla de Bayes, esta información experimental se combina con la información *a priori* anterior a fin de obtener las distribuciones *a posteriori* de cada modelo y total. Ahora los cuatro modelos ya no son equiprobables siendo sus probabilidades 0.2534, 0.2525, 0.2490 y 0.2451, respectivamente. La Figura 4 muestra la distribución *a posteriori* total $p(\omega | \mathbf{y})$ a frecuencia $\omega = 1.16 \text{ rad/s}$, así como la respuesta frecuencial “real”. Nótese como la información experimental ha reducido significativamente la incertidumbre inicial sobre la respuesta frecuencial.

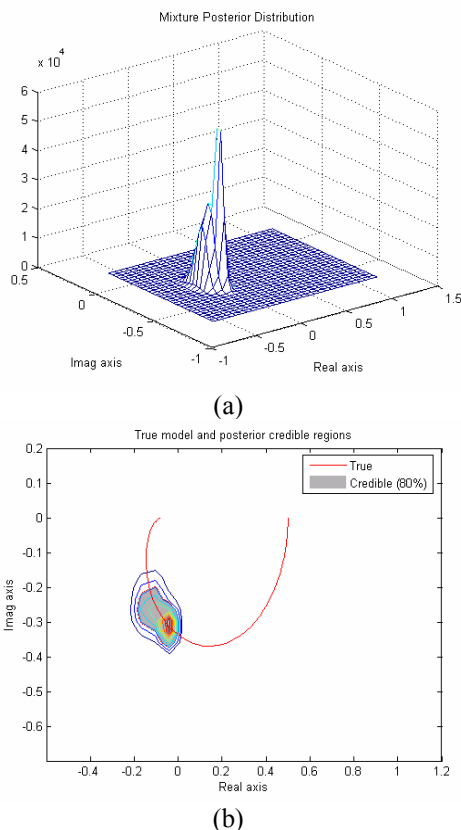


Figura 4: Distribución *a posteriori* total, $p(\omega | \mathbf{y})$

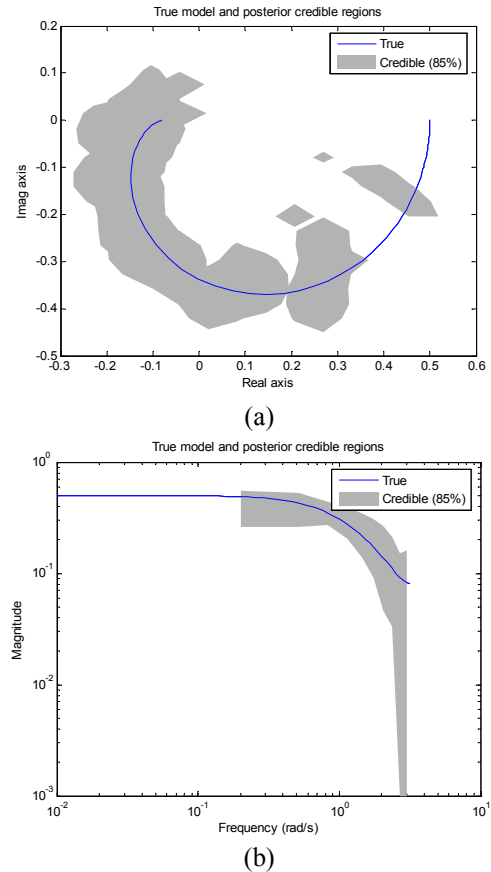


Figura 5: Cotas finales (*a posteriori*) de incertidumbre (a) Nyquist (b) ganancia de Bode

La repetición del procedimiento anterior para 10 frecuencias equiespaciadas entre 0.2 rad/s y 3 rad/s da como resultado las cotas de incertidumbre para un nivel de credibilidad del 85% de la Figura 5. Nótese que, a diferencia de las regiones clásicas, las regiones bayesianas pueden ser disjuntas (ver (a)). Ello nos proporciona información valiosa al respecto de posibles discrepancias entre la información *a priori* y los datos experimentales.

5 CONCLUSIONES

Se ha presentado un enfoque bayesiano para formular y resolver el problema de la identificación de sistemas orientada al control robusto. La principal característica es la necesidad de definir funciones densidad de probabilidad *a priori* sobre el ruido de medida y los modelos candidatos. Si no existe tal información *a priori*, o no es fiable, se eligen distribuciones no informativas, planas comparadas con la función de verosimilitud de las observaciones. En ese caso, las bandas de incertidumbre obtenidas coinciden con las de los actuales métodos estocásticos y de predicción de error. Si la

información *a priori* es informativa, las regiones de incertidumbre resultantes son más pequeñas que las obtenidas a partir de la función de verosimilitud de las observaciones. Por otro lado, es posible reducir aún más el tamaño de las bandas mediante la aplicación reiterada de la regla de Bayes sobre diversos experimentos. Finalmente, el punto de vista bayesiano es altamente unificador y puede ser aplicado a todos los aspectos del problema, desde el diseño óptimo de experimentos hasta la identificación y validación del modelo nominal.

Agradecimientos

Esta comunicación resume algunos resultados de [3]. Queremos agradecer a R.S. Sánchez Peña y M. Sznaier sus comentarios y sugerencias.

Referencias

- [1] Bayes, T., (1763) "An essay towards solving a problem in the doctrine of chances", *Philosophical Transactions of the Royal Society*, 53: 370-418, 1973.
- [2] Chen, J. and Gu, G., (2000) Control-Oriented System Identification. An \mathcal{H}_∞ Approach, John Wiley.
- [3] Fernández-Cantí, R. M., (2006) Bayesian Uncertainty Modelling for Robust Control. A Decision Theory Approach to Robust Identification, Ph. D. Thesis, Dtor.: Herranz, J., UPC.
- [4] Goodwin, G.C., Braslavsky, J.H., and Seron, M. M., (2002) "Non-stationary stochastic embedding for transfer function estimation", *Automatica*, 38: 47-62.
- [5] Green, P., (1995) "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination", *Biometrika*, 82: 711-732.
- [6] Hjalmarsson H. and Gustafsson, F., (1995) "Composite modelling of transfer functions", *IEEE Trans. on Automatic Control*, vol.40.
- [7] Howson C. and Urbach, P., (1989) Scientific Reasoning. The Bayesian Approach. La Salle Open Court.
- [8] Ljung, L., (1999) System Identification - Theory for the User, 2nd ed., Prentice Hall.
- [9] McVinish, R., Braslavsky, J.H., and K. Mengersen K., (2006) "A Bayesian-decision theoretic approach to model error modelling", *Proc. of the 14th IFAC SYSID*.
- [10] Milanese M. and Taragna, M., (2005) " \mathcal{H}_∞ set membership identification: a survey", *Automatica*, 41: 2019-2032.
- [11] Ninness, B. and Gustafsson, F., (1997) "A unifying construction of orthonormal bases for system identification", *IEEE Trans. on Automatic Control*, 42: 515-521.
- [12] Onatski, A. and Williams, N., (2002) "Modelling Model Uncertainty", Working Paper no.169, European Central Bank Working Paper Series, International Seminar on Macroeconomics.
- [13] Peterka, V., (1981) "Bayesian system identification," *Automatica*, 17: 41-53.
- [14] Robert, C. P., (2001) The Bayesian Choice. 2nd ed., Springer Texts in Statistics, Springer Verlag.
- [15] Sánchez Peña, R.S. and Sznaier, M., (1998) Robust Systems Theory and Applications, John Wiley & Sons, Inc.
- [16] Tempo, R., Calafiore, G., and Dabbene, F., (2005) Randomized Algorithms for Analysis and Control of Uncertain Systems, Springer.

UN ENFOQUE PROBABILÍSTICO PARA LA ESTIMACIÓN DEL DESPLAZAMIENTO DE UN ROBOT MÓVIL DOTADO DE VISIÓN ESTÉREO

F.A. Moreno, J.L. Blanco, J. González, V. Arévalo y J.A. Fernández-Madrigal

Departamento de Ingeniería de Sistemas y Automática.

Universidad de Málaga

{famoreno, jlblanco}@isa.uma.es, {jgonzalez, varevalo, jafma}@ctima.uma.es

Resumen

En este artículo se presenta un método probabilístico para estimar el desplazamiento de un robot móvil equipado con un sistema de visión estereoscópico. El método evita el emparejamiento directo entre las marcas (puntos) 3D obtenidas por el sistema estéreo en las distintas posiciones del robot mediante el cálculo de la función de densidad de probabilidad del desplazamiento, que se deriva de la aplicación de la regla de Bayes y del modelado de la función de verosimilitud de las observaciones. Para la obtención de las marcas se resuelve el emparejamiento de las características SIFT extraídas del par de imágenes y se proyectan al espacio tridimensional mediante las matrices de calibración de las cámaras (que son conocidas). El enfoque que proponemos aquí ha sido probado experimentalmente con resultados prometedores.

Palabras Clave: Visión estéreo, marcas visuales, estimación bayesiana, robot móvil.

1 INTRODUCCIÓN

La visión estéreo es una de las más interesantes alternativas para extraer información 3D del entorno en aplicaciones robóticas. Frente a otros métodos como los sistemas de luz estructurada, escáner láser, sónar, etc., la visión estéreo cuenta con ventajas como su relativo bajo coste, cantidad de información proporcionada, versatilidad (aplicación en entornos interiores y exteriores), peso y tamaño del hardware necesario, etc.

En robótica móvil, en concreto, es cada vez más habitual encontrar sistemas estereoscópicos, tanto comerciales como desarrollados a medida, que son empleados para detectar obstáculos, construir mapas, facilitar la interacción con el entorno (humanos u objetos), monitorización remota y/o teleoperación, y estimación de la posición o el desplazamiento del vehículo (ver por ejemplo [7],[8]). En este trabajo se aborda este último problema mediante el emparejamiento de los puntos (marcas) tridimensionales proporcionados por un sistema estereoscópico en dos localizaciones distintas en el espacio.

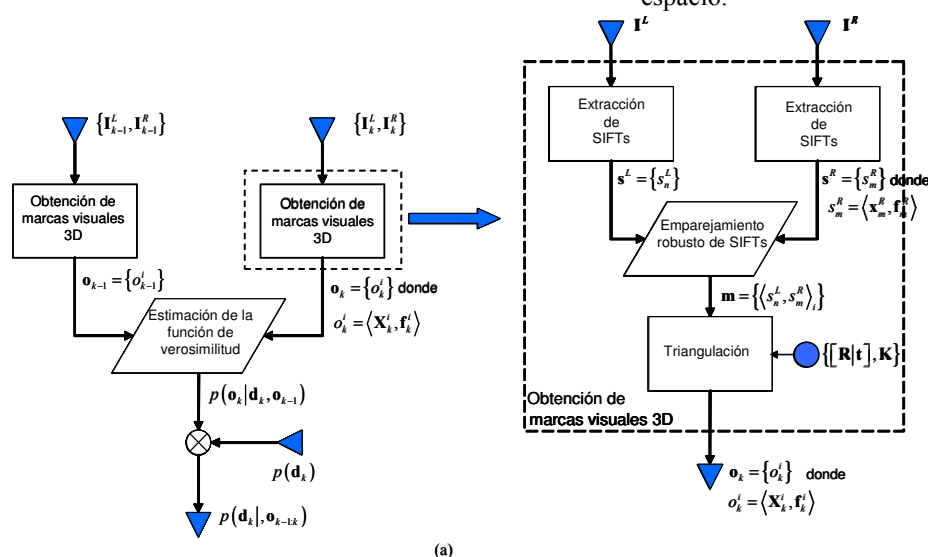


Figura 1: (a) Proceso de estimación del movimiento de un sistema estereoscópico. (b) Imagen de uno de nuestros robots móviles equipado con la cámara estereoscópica BumbleBee®.

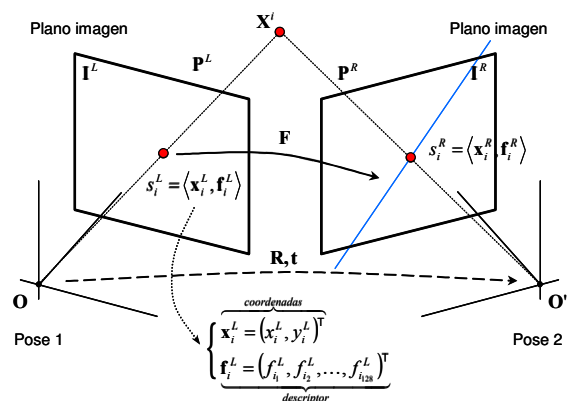


Figura 2: Proceso de extracción de marcas 3D en un sistema de visión estéreo.

Este problema habitualmente se plantea como el cálculo de la transformación plana $\Delta = (\Delta x, \Delta y, \Delta \phi)$ que minimiza el error cuadrático medio entre ambos conjuntos de puntos. De esta clase son, por ejemplo, los algoritmos tipo ICP (*Iterative Closest Point*), que son empleados con éxito con mapas densos de puntos. Estos métodos, aunque fáciles de implementar presentan dos problemas fundamentales:

- 1) Cuando el número de marcas a emparejar es escaso, dejan de ser robustos puesto que son propensos a encontrar mínimos locales que pueden estar lejos de la mejor solución global.
- 2) La solución (i.e. desplazamiento) proporcionada no incluye una medida de su verosimilitud, lo que acarrea serias limitaciones a la hora de fusionar con otros sensores y/o tomar decisiones en base a la confianza en la medida.

El procedimiento propuesto en este trabajo evita estos dos inconvenientes mediante un enfoque bayesiano basado en el modelado de la función de verosimilitud de los puntos observados para todos los posibles desplazamientos del robot.

El proceso completo, esquematizado en la figura 1, ha sido testado con éxito empleando el sistema binocular comercial Bumblebee® [10] en diversos entornos interiores (pasillos, despachos, laboratorios, etc.). Los resultados demuestran que la precisión alcanzada es mayor que la proporcionada por un método de mínimos cuadrados, y suficiente para muchas aplicaciones de posicionamiento.

A continuación se describe el proceso de obtención de marcas visuales 3D a partir del sistema de visión estéreo. En la sección 3 se describe y formula la función de densidad de probabilidad del desplazamiento, a partir de la función de verosimilitud de la observación. La sección 4 está dedicada a presentar las pruebas realizadas y analizar

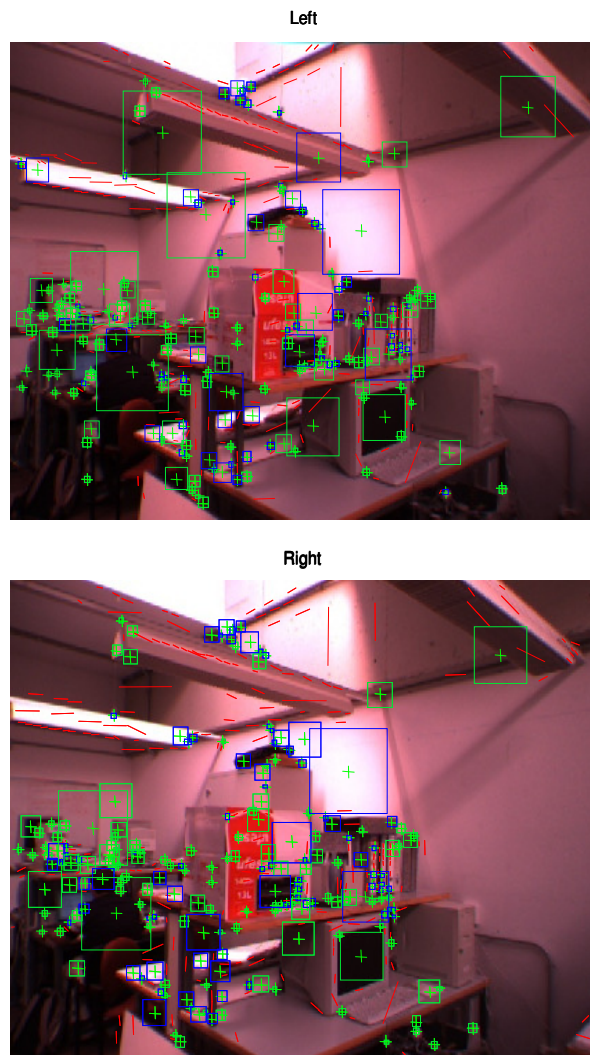


Figura 3: Características localizadas por el detector de Lowe en un par de imágenes estereoscópicas.

sus resultados. Finalmente se presentan las conclusiones y futuras líneas de extensión de este trabajo.

2 OBTENCIÓN DE PUNTOS 3D EN UN SISTEMA DE VISIÓN ESTÉREO

En esta sección se aborda la obtención de marcas 3D a partir de un par de imágenes proporcionadas por un sistema de visión estéreo calibrado. El proceso de extracción consta de las siguientes etapas: 1) identificación de características en ambas imágenes, 2) emparejamiento robusto de las características seleccionadas (y eliminación de pares espurios) y finalmente, 3) la generación, a partir de los pares identificados, de marcas 3D mediante triangulación. La figura 2 ilustra gráficamente el proceso de identificación, emparejamiento y triangulación.

A continuación se describen pormenorizadamente cada una de estas etapas.

2.1 EXTRACCIÓN DE CARACTERÍSTICAS

La primera etapa del proceso de obtención de marcas 3D consiste en identificar características en el par de imágenes estéreo. Una característica es cualquier objeto distintivo o representativo que puede ser detectado en la imagen: un borde, región, contorno, esquina, etc. Si nos centramos en la búsqueda de esquinas (puntos de máxima curvatura), son muchas las técnicas propuestas en la literatura para tal fin. Algunas de ellas explotan las derivadas de primer orden (como el conocido detector de Harris [1]) o de segundo orden [3] de la imagen. Otros métodos van más allá e incorporan descriptores a las esquinas localizadas con objeto de dotarles de invarianza frente a, por ejemplo, cambios de escala [9], o incluso distorsiones afines [6], incrementando así la robustez y facilitando su posterior emparejamiento.

El detector empleado en este trabajo (denominado detector de Lowe [4]), está dentro de esta última categoría. Este método aborda la detección mediante una búsqueda de extremos (máximos y mínimos locales) en un espacio de escalas construido a partir de Diferencias de Gaussianas (DoG). Las esquinas (de coordenadas $\mathbf{x} = (x, y)^T$) localizadas con este procedimiento son caracterizadas mediante un descriptor $\mathbf{f} = (f_1, \dots, f_8)^T$ basado en información local del gradiente que las dota de invarianza a transformaciones afines y cambios de iluminación; el par $s = \langle \mathbf{x}, \mathbf{f} \rangle$ se denomina característica SIFT¹. La figura 3 muestra las características identificadas en un par de imágenes estéreo como las utilizadas en este trabajo.

2.2 OBTENCIÓN DE PARES

Una vez finalizada la selección de características en el par de imágenes estéreo, la siguiente etapa del proceso aborda su emparejamiento. A diferencia de otras técnicas tradicionales que explotan la similitud radiométrica de los pares candidatos (y de su entorno) mediante correlación cruzada normalizada NCC, suma de diferencias al cuadrado SSD, etc., el procedimiento utilizado en este trabajo establece las correspondencias de acuerdo con la distancia euclídea que separa sus descriptores (que son menos sensibles a los cambios de iluminación y a las posibles distorsiones geométricas que pudieran presentar las imágenes). Por otro lado, con objeto de aportar robustez al proceso, descartamos aquellos pares (espurios) cuyas coordenadas no son consistentes con la geometría epipolar que relaciona ambas imágenes.

¹ De los términos ingleses "Scale-Invariant Feature Transform".

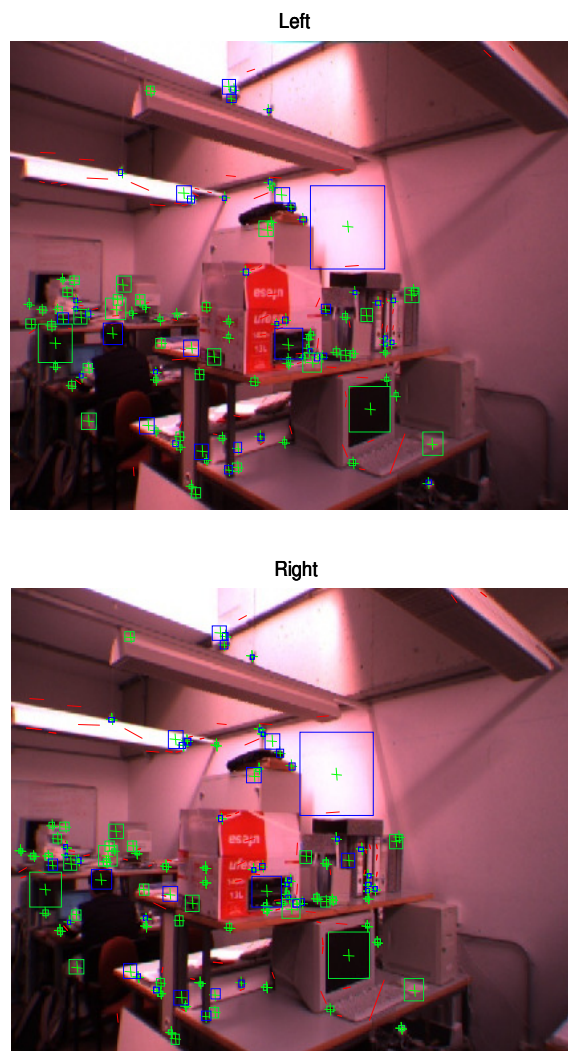


Figura 4: Pares de correspondencias establecidos mediante el procedimiento descrito en la sección 2.2.

Sean, por tanto, dos conjuntos de SIFTs $\mathbf{s}^L = \{s_i^L, i = 1, \dots, n\}$ y $\mathbf{s}^R = \{s_j^R, j = 1, \dots, m\}$ identificados en la imagen izquierda \mathbf{I}^L y derecha \mathbf{I}^R , respectivamente. El proceso de emparejamiento consiste en identificar de forma unívoca los pares $\langle s_i^L, s_j^R \rangle$ tales que:

- a) la norma euclídea de sus descriptores $\|\mathbf{f}_i^L - \mathbf{f}_j^R\|$ esté por debajo de un umbral y sea mínima,
- b) sus coordenadas verifiquen la restricción epipolar $(\mathbf{x}_j^R)^T \mathbf{F} \mathbf{x}_i^L \approx 0$, esto es, la distancia de \mathbf{x}_j^R a su correspondiente línea epipolar $\mathbf{F} \mathbf{x}_i^L$ sea muy pequeña (por ejemplo, menor de un píxel),
- c) y finalmente, su disparidad sea superior a tres píxeles. Con esta actuación evitamos configuraciones que conlleven importantes errores en el proceso de triangulación (haces casi paralelos).

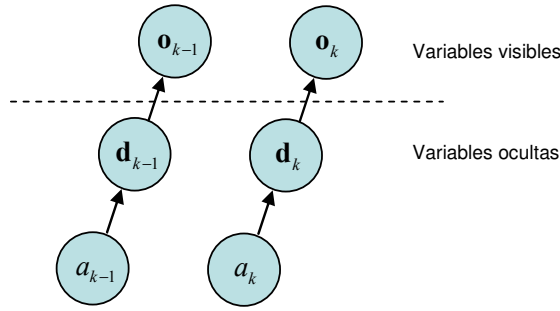


Figura 5: Modelo gráfico del problema en el que se observan las dependencias entre las variables involucradas. Las variables a_k , o_k y d_k representan las acciones, observaciones y desplazamientos en el instante k , respectivamente.

Obsérvese que, puesto que el par de cámaras estéreo se supone calibrado, están disponibles la matriz de calibración \mathbf{K} y la disposición relativa de las cámaras \mathbf{R} y \mathbf{t} [2]. A partir de ellas se calcula la matriz fundamental \mathbf{F} requerida en la condición b) anterior mediante:

$$\mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \quad (1)$$

donde $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$ es la matriz esencial [2].

La figura 4 muestra los emparejamientos encontrados por el procedimiento expuesto para las características de las imágenes de la figura 3.

2.3 TRIANGULACIÓN

Una vez establecidas de manera robusta las correspondencias entre los SIFTs identificados en el par de imágenes estéreo, la siguiente etapa del proceso consiste en determinar su localización tridimensional en el mundo real.

Sean $\mathbf{P}^L = [\mathbf{I} | 0]$ y $\mathbf{P}^R = [\mathbf{R} | \mathbf{t}]$ las matrices de las cámaras izquierda y derecha respectivamente, y $\mathbf{x}^L = \mathbf{K} \mathbf{P}^L \mathbf{X}$ y $\mathbf{x}^R = \mathbf{K} \mathbf{P}^R \mathbf{X}$ las proyecciones del punto tridimensional \mathbf{X} sobre los planos imagen de ambas cámaras. Puesto que ambas proyecciones verifican $\mathbf{x}^L \times (\mathbf{K} \mathbf{P}^L \mathbf{X}) = 0$ (análogamente para \mathbf{x}^R), el proceso de triangulación se puede expresar linealmente como sigue:

$$\mathbf{A} \mathbf{X} = 0 \quad (2)$$

con

$$\mathbf{A} = \begin{pmatrix} x^L \mathbf{p}^{L3T} - \mathbf{p}^{L1T} \\ y^L \mathbf{p}^{L3T} - \mathbf{p}^{L2T} \\ x^R \mathbf{p}^{R3T} - \mathbf{p}^{R1T} \\ y^R \mathbf{p}^{R3T} - \mathbf{p}^{R2T} \end{pmatrix} \quad (3)$$

donde \mathbf{p}^{LiT} es la i -ésima fila de \mathbf{P}^L transpuesta (análogamente para \mathbf{P}^R).

Resolviendo (2) (mediante, por ejemplo, la descomposición de (3) en valores singulares) para todos y cada uno de los pares de SIFTs identificados en el par de imágenes estéreo, obtenemos sus correspondientes coordenadas 3D. A cada uno de estos puntos tridimensionales se le asocia el descriptor SIFT de una de sus proyecciones en las imágenes² (por ejemplo, en la izquierda), formando, de esta manera, un conjunto de pares $\langle \mathbf{X}_k^i, \mathbf{f}_k^i \rangle$ al que en adelante denominaremos observación.

3 ESTIMACION DE LA FUNCIÓN DE DENSIDAD DE PROBABILIDAD DEL DESPLAZAMIENTO

En esta sección abordamos el cálculo de la densidad de probabilidad del desplazamiento incremental (\mathbf{d}_k) en el instante de tiempo k condicionado a las dos observaciones anterior y actual (\mathbf{o}_{k-1} y \mathbf{o}_k):

$$p(\mathbf{d}_k | \mathbf{o}_{k-1}, \mathbf{o}_k) \quad (4)$$

Las observaciones están compuestas por un conjunto de marcas tridimensionales obtenidas mediante triangulación de las características SIFT.

Siguiendo un enfoque bayesiano, la función (4) puede descomponerse en un término de estimación *a priori* y un término de verosimilitud:

$$p(\mathbf{d}_k | \mathbf{o}_{k-1}, \mathbf{o}_k) = \underbrace{\eta}_{\text{Estimación a priori}} \cdot \underbrace{p(\mathbf{o}_k | \mathbf{d}_k, \mathbf{o}_{k-1})}_{\text{Función de verosimilitud}} \quad (5)$$

donde $\eta = 1/p(\mathbf{o}_k)$, representa la inversa de la probabilidad de la observación en el instante k , la cual es una constante cuyo valor no es significativo para nuestros propósitos.

A partir del modelo gráfico del problema (figura 5) se observa que \mathbf{d}_k y \mathbf{o}_{k-1} son variables aleatorias independientes, por lo que en la estimación *a priori* en (5) la observación \mathbf{o}_{k-1} no aporta información a la distribución del desplazamiento \mathbf{d}_k . Además, suponiendo que no tenemos información alguna sobre el desplazamiento (al margen de \mathbf{o}_k), asumimos que esta densidad sigue una distribución uniforme:

$$p(\mathbf{d}_k | \mathbf{o}_{k-1}) = p(\mathbf{d}_k) = \eta' \quad (6)$$

² Los descriptores de las proyecciones deben ser muy similares al ser puntos correspondientes, por lo que cualquiera de ellas servirá como distintivo para el punto tridimensional.

De esta forma, la expresión (5) se reduce a:

$$p(\mathbf{d}_k | \mathbf{o}_{k-1}, \mathbf{o}_k) = \eta \cdot \eta' \cdot p(\mathbf{o}_k | \mathbf{d}_k, \mathbf{o}_{k-1}) \quad (7)$$

donde el producto $\eta \cdot \eta' = 1$ ya que las densidades a la izquierda y derecha de la expresión (7) deben tener integral unidad. Por lo tanto, en nuestro caso, la función de verosimilitud tiene el mismo valor numérico que la densidad de probabilidad del desplazamiento incremental.

A continuación exponemos nuestra solución al problema del cálculo de dicha función. Asumiendo independencia en el proceso de detección de cada una de las características (\mathbf{o}_k^i), podemos factorizar la función de verosimilitud como:

$$p(\mathbf{o}_k | \mathbf{d}_k, \mathbf{o}_{k-1}) = \prod_i p(\mathbf{o}_k^i | \mathbf{d}_k, \mathbf{o}_{k-1}) \quad (8)$$

En este trabajo proponemos aproximar la distribución de probabilidad de observar una característica, dada una observación anterior \mathbf{o}_{k-1} y un desplazamiento \mathbf{d}_k , mediante una extensión de un método utilizado anteriormente para sensores láser radiales, llamado *campo de verosimilitud (likelihood field)* [11]. Este método considera únicamente los puntos detectados en cada medida (los puntos terminales de cada rayo), y calcula su probabilidad asumiendo que la distancia entre estos puntos y un mapa se corresponde con un error de medida gaussiano. En nuestro caso, las medidas, además de la posición 3D del punto observado, incorporan un descriptor que vamos a explotar para resolver la incertidumbre en las correspondencias.

Sea c_{ij} una variable aleatoria que indica la probabilidad de que el punto 3D observado en el instante k (\mathbf{o}_k^i) se corresponda con el punto \mathbf{o}_{k-1}^j . En este trabajo se define la probabilidad de una correspondencia c_{ij} condicionada a un desplazamiento \mathbf{d}_k y la observación en el instante anterior como:

$$p(c_{ij} | \mathbf{d}_k, \mathbf{o}_{k-1}) \propto e^{-\frac{1}{2} \frac{\|\mathbf{f}_k^i - \mathbf{f}_{k-1}^j\|^2}{\sigma_F^2}} \quad (9)$$

donde \mathbf{f}_k^i y \mathbf{f}_{k-1}^j son los descriptores de los puntos 3D i y j en los instantes k y $k-1$, respectivamente, y el parámetro σ_F modela los errores en el cálculo de descriptores de un mismo punto visto desde distintas posiciones. Esta función asigna una probabilidad alta a la correspondencia entre pares de características con descriptores similares, lo que se estima mediante la distancia euclídea entre descriptores. Normalmente esta función sólo tendrá un valor significativo para la correspondencia correcta de la característica.

Por otro lado, para calcular la probabilidad de una observación i en el instante k condicionada al desplazamiento en el mismo instante, a la observación en el instante anterior y a que se produzca la correspondencia c_{ij} , utilizamos la misma aproximación que en el método del campo de verosimilitud:

$$p(\mathbf{o}_k^i | \mathbf{d}_k, \mathbf{o}_{k-1}, c_{ij}) \propto e^{-\frac{1}{2} \frac{\|\mathbf{X}_k^i(\mathbf{d}_k) - \mathbf{X}_{k-1}^j\|^2}{\sigma_D^2}} \quad (10)$$

donde $\mathbf{X}_k^i(\mathbf{d}_k)$ y \mathbf{X}_{k-1}^j son las posiciones tridimensionales de los puntos i y j en los instantes k y $k-1$, respectivamente y σ_D modela el error del proceso de detección de las características en las imágenes. Nótese que la posición de las características en el momento k dependen del desplazamiento \mathbf{d}_k .

En este artículo, proponemos la siguiente aproximación de la función de verosimilitud para una observación i :

$$p(\mathbf{o}_k^i | \mathbf{d}_k, \mathbf{o}_{k-1}) = \sum_j \underbrace{p(\mathbf{o}_k^i | \mathbf{d}_k, \mathbf{o}_{k-1}, c_{ij})}_{\text{Término de distancia}} \cdot \underbrace{p(c_{ij} | \mathbf{d}_k, \mathbf{o}_{k-1})}_{\text{Término de correspondencia}} \quad (11)$$

donde se ha aplicado el teorema de la probabilidad total.

Para ilustrar el tipo de resultados obtenidos con esta función de verosimilitud, en la figura 6 se muestran los valores estimados, para un entorno de interior, en el caso de un desplazamiento real de 0.665m en el eje x y 0.049m en el eje y . Estos valores han sido estimados mediante alineamiento de *scans* de un escáner láser radial, que se ha tomado como referencia precisa del desplazamiento. Se puede ver que el máximo de la verosimilitud está cerca de estos valores.

4 RESULTADOS EXPERIMENTALES

En esta sección exponemos los resultados obtenidos a partir de las imágenes estéreo capturadas durante la navegación de nuestro robot móvil SANCHO, equipado con una cámara BumbleBee® (figura 1(b)). El experimento descrito a continuación valida nuestro método de aproximación de la función de verosimilitud de imágenes estéreo.

El experimento consiste en estimar la posición del robot de manera incremental durante su navegación a lo largo de un pasillo situado en la E.T.S.I. Informática de Málaga. Aprovechando que el robot también está equipado con un sensor láser radial SICK LMS220, utilizaremos como *ground truth* de

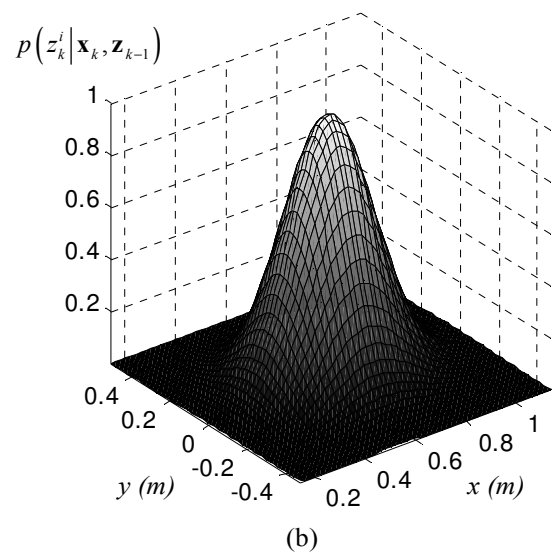
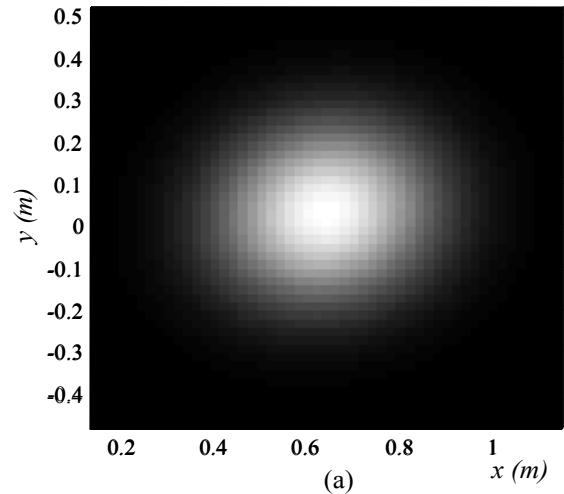
los desplazamientos incrementales el resultado de un algoritmo de *scan-matching* (ICP) entre *scans* sucesivos, cuyos resultados tienen un error residual estimado inferior a 1 cm [5].

En cuanto a la detección de características SIFT y el emparejamiento entre cada par de imágenes estereó, en la figura 7(a) se muestra el número de características detectadas en cada una de las imágenes, así como el número de correspondencias encontradas entre ellas a lo largo de la navegación. Cabe destacar que el número medio de correspondencias encontradas está en torno a 70, lo que representa aproximadamente un 25% de las marcas encontradas en cada una de las imágenes. Como ejemplo del proceso de triangulación para la obtención de marcas en 3D a partir de correspondencias, en la figura 7(d) mostramos un conjunto de éstas proyectadas sobre un plano horizontal. Estas marcas se corresponden con las características detectadas en las imágenes 7(b)-(c).

En la figura 8(a) se puede ver la trayectoria estimada por nuestro método al aplicar el desplazamiento incremental estimado en cada paso a la posición dada por el *ground truth* en el paso anterior. Podemos resaltar el pequeño error cometido con respecto a la estimación realizada a partir del sensor láser. La precisión de nuestro método se aprecia más claramente en la ampliación de un tramo de la trayectoria que se muestra en la figura 8(b). En ella se representa con un punto azul la media de la función de verosimilitud para sucesivos instantes de tiempo, junto con la elipse que cubre el área correspondiente al intervalo de confianza del 95%. Se puede observar cómo en la mayoría de los casos la media del desplazamiento estimado resulta muy próxima a la estimación del ICP (representada en la figura por un punto rojo). La distribución de la distancia entre ambas estimaciones se muestra en forma de histograma en la figura 8(c). Esta distancia se puede considerar un indicador de la calidad de nuestro método. Para este experimento se obtiene un valor medio de 7.73 cm. Teniendo en cuenta que los desplazamientos son del orden de 80cm, este resultado indica que nuestro método es relativamente preciso.

5 CONCLUSIONES

En este trabajo se ha abordado el problema de estimar el desplazamiento diferencial de un sistema de visión estereó de forma probabilística. Para ello hemos utilizado un método de estimación de la función de verosimilitud anteriormente empleado solamente con *scans* láser en 2D. En este trabajo hemos extendido el ámbito de aplicación de este método a características tridimensionales.



Los resultados experimentales obtenidos indican que las estimaciones dadas por nuestro método cometen un error relativamente bajo, concretamente, obtenemos un error medio de 7.73 cm. A pesar de que este error es claramente superior al alcanzable a partir de un sensor láser radial, mediante visión se pueden detectar características en una parte mayor del espacio alrededor del robot. Por ejemplo, si el robot está rodeado de personas la mayoría de las lecturas láser serán inválidas para localización. Sin embargo, mediante visión existe más probabilidad de detectar marcas útiles para localización, por ejemplo: techos, paredes a distintas alturas, etc.

El método presentado ofrece importantes oportunidades de aplicación en los campos de localización y construcción de mapas probabilísticos, ya que permite su integración de manera directa en aproximaciones secuenciales de Monte-Carlo (filtros de partículas).

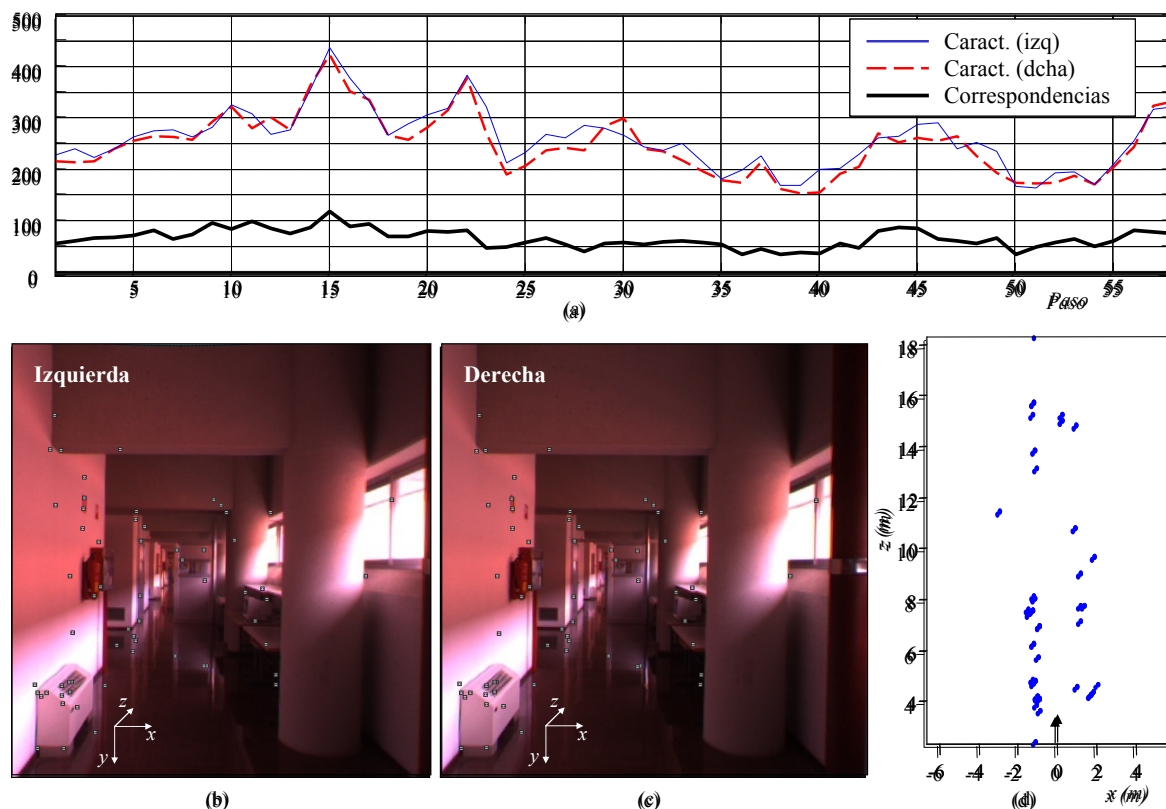


Figura 7: (a) Número de características detectadas en cada una de las imágenes estéreo a lo largo de la navegación, junto con el número de correspondencias encontradas. (b)-(c) Imágenes capturadas por las cámaras izquierda y derecha, respectivamente, en un momento dado del experimento. En las figuras se resaltan las características para las que se encuentran correspondencias válidas. (d) Vista superior de la proyección en 3D de dichas correspondencias sobre la planta del entorno. La flecha indica la orientación del robot en el momento en el que se tomaron las imágenes.

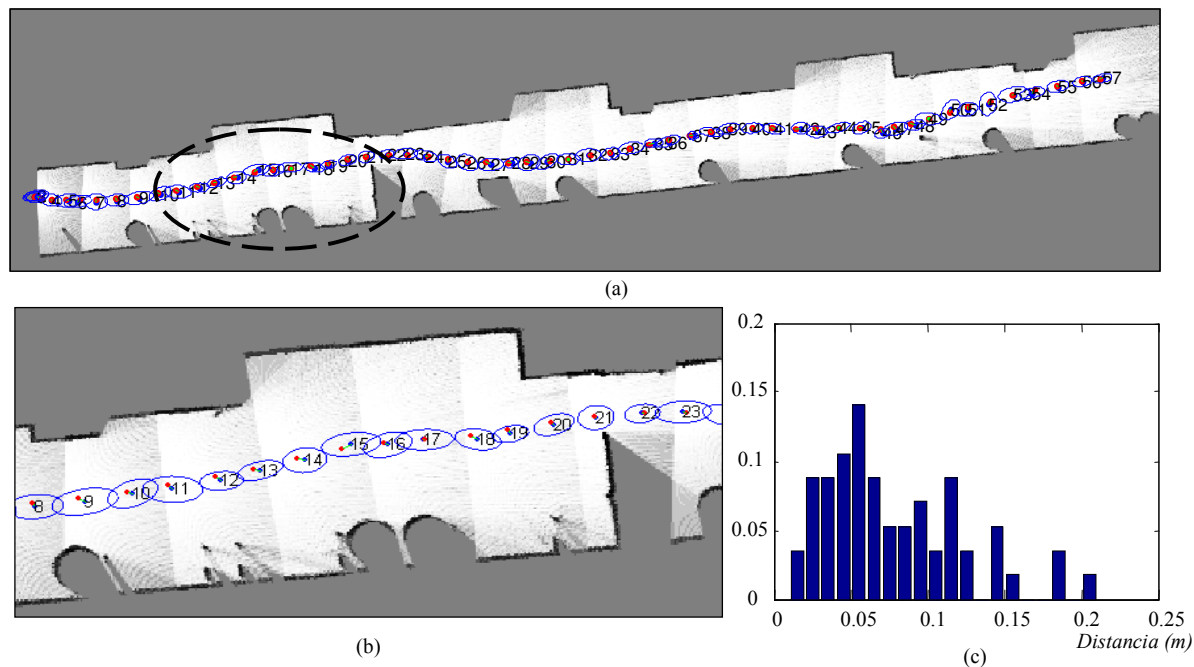


Figura 8: (a) Trayectoria estimada a partir de nuestro método para calcular la función de verosimilitud de desplazamientos incrementales. Por claridad, se muestran las elipses de incertidumbre de la estimación superpuestas a un mapa del entorno. (b) Vista ampliada del tramo destacado en la figura (a), donde se aprecia la precisión de nuestro método comparado con la estimación a partir del alineamiento de *scans* láser (ICP). (c) Distribución de la distancia entre la estimación de nuestro método y la del ICP.

Referencias

- [1] Harris, C.J. y Stephens, M. (1988). "A combined corner and edge detector". In *Proceedings of 4th Alvey Vision Conference*, pp. 147-151. Manchester.
- [2] Hartley, R.I. y Zisserman, A. (2000). "Multiple view geometry in computer vision". *Cambridge University Press*.
- [3] Kitchen, L. y Rosenfeld, A. (1982). "Gray-level corner detection". *Pattern Recognition Letters*, 1, pp. 95-102.
- [4] Lowe, D.G. (2003). "Distinctive image features from scale-invariant keypoints". *Technical Report*, Computer Science Department (University of British Columbia).
- [5] Martínez, J.L., González, J., Morales, J., Mandow, A. y García-Cerezo, A. (2006). "Genetic and ICP laser point matching for 2D mobile robot motion estimation". In *Journal of Field Robotics*, vol.23.
- [6] Mikolajczyk, K. y Schmid, C. (2002). "An affine invariant interest point detector". In *European Conference on Computer Vision (ECCV)*, pp. 128-142. Copenhagen, Denmark.
- [7] Murray D. y Jennings C. (1997). "Stereo vision based mapping and navigation for mobile robots". In *International Conference on Robotics and Automation*, pp. 1694-1699. Albuquerque, Nuevo Mexico.
- [8] Murray D. y Little J.J. (2000). "Using Real-Time Stereo Vision for Mobile Robot Navigation". In *Autonomous Robots* 8, pp. 161-171
- [9] Schmid, C. y Mohr, R. (1997). "Local gray-value invariants for image retrieval". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), pp. 530-534.
- [10] Sitio Web de la cámara BumbleBee de PGR, www.ptgrey.com/products/bumblebee/
- [11] Thrun, S., Burgard, W. y Fox, D. (2005). "Probabilistic Robotics". *The MIT Press*.

OPTIMIZACIÓN DINÁMICA EN ECOSIMPRO®. APLICACIÓN A LA ESTIMACIÓN DE PARÁMETROS Y VALIDACIÓN DE MODELOS

Elena Gómez

Centro de Tecnología Azucarera- Univ. de Valladolid, elenags@cta.uva.es

César de Prada

Dpto. Ingeniería de Sistemas y Automática - Univ. de Valladolid, prada@autom.uva.es

Resumen

Las técnicas de optimización dinámica se aplican para resolver problemas de gran interés: ajuste de los parámetros desconocidos de un modelo dinámico, validación de modelos, control predictivo basado en modelo. En el entorno del lenguaje de simulación EcosimPro®, se evaluó en primer lugar el funcionamiento de una herramienta de optimización y después se estudió de forma sistemática el problema de calibración y validación de modelos.

Palabras Clave: Optimización dinámica, estimación de parámetros, análisis de sensibilidad, identificabilidad, validación de modelos.

1 OPTIMIZACIÓN DINÁMICA

Son muchas las aplicaciones de la optimización en el campo de la ingeniería de procesos, cubriendo desde problemas de diseño a técnicas de control avanzado como el control predictivo (MBPC). En muchos casos, como ocurre a menudo en problemas de diseño, se trata de decisiones de tipo estático, donde las variables de decisión y el proceso considerado no evolucionan en el tiempo, sino que corresponden a una determinada situación de equilibrio. Por el contrario, en otros casos, las variables involucradas y el proceso evolucionan en el tiempo dando lugar a problemas de optimización dinámica.

En los problemas de optimización de sistemas dinámicos se desea optimizar una función objetivo $J(\mathbf{u}(t), \mathbf{x}(t))$ que es función de unas variables de decisión $\mathbf{u}(t)$ y de las variables de estado $\mathbf{x}(t)$ del proceso, que a su vez evolucionan en el tiempo. Las restricciones de un problema de optimización dinámica incluyen las ecuaciones diferenciales del modelo junto a otras como los límites inferior y superior de las variables de decisión, u otras particulares de cada problema. Cuando las variables

de decisión son de tipo real, los problemas resultantes se denominan de programación no-lineal dinámica.

Un problema general de este tipo con restricciones de igualdad y desigualdad se formula habitualmente del siguiente modo (1):

$$\begin{aligned} &\text{minimizar} && J(\mathbf{u}, \mathbf{x}) && (1) \\ &\text{respecto a} && \mathbf{u} \\ &\text{sujeto a:} && \\ &&& g_m(\mathbf{x}, \mathbf{u}) \geq 0 && m = 1, 2, \dots, m \\ &&& h_k(\mathbf{x}, \mathbf{u}) = 0 && k = 1, 2, \dots, k \\ &&& \mathbf{u}_{inferior} \leq \mathbf{u} \leq \mathbf{u}_{superior} \\ &&& \frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ &&& \hat{\mathbf{y}}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \end{aligned}$$

donde $\mathbf{x}(t)$ son los estados del sistema, $\hat{\mathbf{y}}(t)$ son las predicciones del modelo para las respuestas medibles del sistema y $\mathbf{u}(t)$ es un vector de variables de decisión que puede variar en función del tipo de problema considerado.

Hay varias técnicas posibles para su resolución que tratan de reformular el problema en términos de uno de programación no-lineal (NLP) y pueden clasificarse dentro de dos grandes familias: Los llamados métodos simultáneos, que reformulan la parte dinámica de las restricciones convirtiéndola en un conjunto de ecuaciones algebraicas, bien mediante métodos de colocación o fórmulas de integración numérica, y los llamados métodos sucesivos, que se basan en la integración de las ecuaciones dinámicas y serán los considerados en este trabajo.

Así pues, la función objetivo a minimizar se calculará para unos valores determinados de las variables de decisión mediante la simulación del modelo correspondiente desarrollado en EcosimPro®, que es el entorno de modelado empleado.

Las variables de decisión a optimizar y la función objetivo son distintas dependiendo del tipo de problema de optimización dinámica de que se trate.

En este trabajo se considerará un caso de estimación de parámetros de un modelo, el cual consiste en hallar los parámetros desconocidos de un modelo dinámico para que su respuesta se ajuste lo más posible a los datos experimentales del proceso real. Resulta de gran interés dado que por lo general siempre existen parámetros que no pueden conocerse previamente de forma sencilla (bibliografía, experimentación específica) y que necesitan ser estimados, de modo que las variables de decisión son los valores de los parámetros desconocidos de un modelo y la función objetivo mide la distancia entre las respuestas del modelo y unos datos experimentales.

1.1 OBJETIVOS

Para la estimación mediante optimización dinámica de los parámetros desconocidos y la validación, se empleó el modelo correspondiente a un proceso muy sencillo consistente en un depósito al que llega un flujo de entrada y del que sale un flujo por gravedad, que dispone de agitación y una resistencia eléctrica calefactora (Figura 1). Las hipótesis consideradas son las de mezcla perfecta, propiedades físicas constantes e inercia térmica del recipiente despreciable. El modelo se formula a partir de las ecuaciones diferenciales de los balances de materia y energía (2 y 3). Las salidas medidas se corresponden con los estados del sistema: nivel h , temperatura T . Los supuestos parámetros desconocidos θ a estimar son 4: k factor de fricción de la tubería de salida (θ_1), U_{amb} coeficiente de pérdida de calor al ambiente (θ_2), A superficie del depósito (θ_3), R resistencia eléctrica (θ_4). Las incertidumbres se supusieron de $\pm 25\%$.

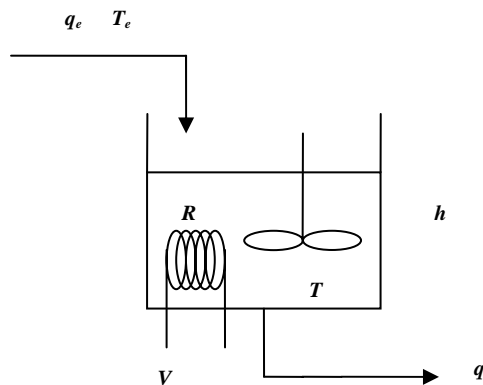


Figura 1: Esquema del modelo del depósito.

Siendo q_e y T_e el caudal y temperatura de la alimentación, ρ y C_p la densidad y el calor específico del líquido, T_{amb} la temperatura externa del ambiente y V la tensión aplicada a la resistencia eléctrica:

$$A \cdot \frac{dh}{dt} = q_e - k\sqrt{h} \quad (2)$$

$$A \cdot h \cdot \rho \cdot C_p \cdot \frac{dT}{dt} = q_e \cdot \rho \cdot C_p \cdot (T_e - T) + \frac{V^2}{R} - U_{amb} \cdot (T - T_{amb}) \quad (3)$$

Con el modelo del depósito se analizaron las distintas técnicas empleadas. Los supuestos datos reales del proceso se obtuvieron mediante simulación, incluyéndose en las señales ruido generado a partir de un modelo ARMA (auto-regresivo y de media móvil).

2 ESTIMACIÓN DE PARÁMETROS

Antes de efectuar la estimación de parámetros mediante técnicas de optimización dinámica es necesario realizar un análisis para determinar cuáles de ellos es conveniente estimar. Para ello se utilizó un procedimiento sistemático, que en primer lugar analiza las sensibilidades del modelo respecto a los parámetros desconocidos para determinar cuáles son los más importantes, en segundo lugar evalúa la identificabilidad de distintos subconjuntos de parámetros (que incluyen aquellos importantes individualmente de acuerdo con la sensibilidad) y, por último, selecciona uno o varios subconjuntos de parámetros a estimar.

2.1 RESOLUCIÓN POR OPTIMIZACIÓN

El planteamiento y la forma de resolver un problema de estimación de parámetros en términos de optimización considera que para cada valor del vector de parámetros θ (variables de decisión) el modelo proporciona una predicción de la respuesta del sistema $\hat{y}(\theta)$ en un experimento determinado. Se toman muestras de los datos de entradas $u(t)$ y salidas $y(t)$ del sistema real sobre un período de tiempo $t = 1, \dots, N$. Al modelo se le aplica la misma secuencia de variables manipuladas $u(t)$ que al sistema. Para cada instante de tiempo t , el error de predicción $e(t)$ (4) es una medida de la bondad del modelo:

$$e(\theta, t) = \hat{y}(\theta, t) - y(t) \quad (4)$$

y se trata de encontrar los valores de los parámetros θ que minimizan los errores de predicción a lo largo de un experimento. Esto se puede formular como un problema de optimización según (5):

$$\min_{\theta} J = \sum_i \sum_{t=1}^N \gamma_i [e_i(t)]^2 = \sum_i \sum_{t=1}^N \gamma_i \cdot [\hat{y}_i(\mathbf{u}, \theta, t) - y_i(t)]^2 \quad (5)$$

sujeto a las restricciones (6) y (7):

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta, t) \quad (6)$$

$$\hat{y}(t) = g(x(t), u(t), \theta, t)$$

$$\theta_{inferior} \leq \theta \leq \theta_{superior} \quad (7)$$

donde J es la función objetivo que evalúa el ajuste del modelo a los datos experimentales reales para el vector de parámetros θ , y el sumatorio se extiende a los instantes de tiempo muestreados t y a cada una de las salidas medidas i del sistema. Aunque el problema es multiobjetivo, lo mejor suele ser ajustar conjuntamente todas las salidas medidas. Los factores de peso γ_i permiten ponderar de forma distinta el ajuste en función de la precisión que se quiera en cada salida; sirven además para normalizar las salidas y que sean homogéneas.

Así pues, la función objetivo se formula a partir de un criterio de mínimos cuadrados ponderados. Los pesos γ_i suelen tomarse como los inversos de la varianza del ruido de las salidas medidas $\gamma_i = 1/\sigma_i^2$. De este modo se consigue que los subtotaes J_i de la función objetivo correspondientes a cada salida i tengan valores comparables en el óptimo, y por tanto que todas las salidas tengan la misma importancia relativa en el ajuste.

En una formulación (8) equivalente a (5):

$$\min_{\theta} J = \sum_{t=1}^N [\hat{y}(\mathbf{u}, \theta, t) - \mathbf{y}(t)]^T \cdot \mathbf{Q} \cdot [\hat{y}(\mathbf{u}, \theta, t) - \mathbf{y}(t)] \quad (8)$$

donde \mathbf{Q} es una matriz diagonal de pesos, igual a la inversa de la matriz de covarianzas del ruido de las medidas según lo indicado anteriormente.

El ruido de medida o debido a perturbaciones aleatorias puede caracterizarse con datos de las salidas de un experimento manteniendo las variables manipuladas constantes. La varianza del ruido del proceso para la salida i se estima según (9), donde \bar{y}_i es el valor medio de la salida i :

$$\sigma_i^2 = \frac{1}{N-1} \sum_{t=1}^N (y_i - \bar{y}_i)^2 \quad (9)$$

Por último se introducirá la Matriz de Información de Fisher (FIM), dado que se utilizará posteriormente. La FIM mide las sensibilidades del modelo respecto del vector de parámetros y se define según (10):

$$\text{FIM} = \left[\sum_{t=1}^N \left(\frac{\partial \hat{y}(t)}{\partial \theta} \right)^T \cdot \mathbf{Q} \cdot \left(\frac{\partial \hat{y}(t)}{\partial \theta} \right) \right] \quad (10)$$

Aunque no se aplicó en el presente trabajo, dicha matriz FIM es la base para el diseño óptimo de experimentos, dado que al maximizar la FIM –o alguna norma o función asociada– se maximiza la información extraída del sistema.

2.2 EXPERIMENTACIÓN

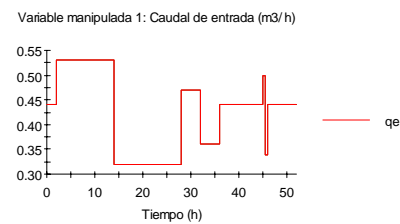


Figura 2: Secuencia de variable manipulada 1.

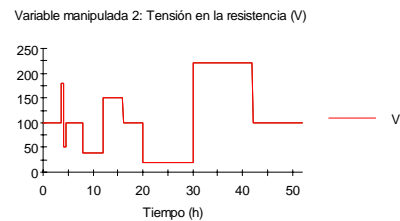


Figura 3: Secuencia de variable manipulada 2.

En la toma de datos experimentales del proceso real habrán de tomarse al menos dos conjuntos, uno para la calibración y otro para la validación del modelo. De acuerdo con las consideraciones habituales en experimentación, se eligió un período de muestreo adecuado; amplitud y frecuencia de las entradas convenientes para excitar las dinámicas fundamentales del sistema; se especificaron entradas no correlacionadas y condiciones de operación del proceso de interés. En las Figuras 2 y 3 puede verse un ejemplo de secuencia de entradas a aplicar al proceso.

2.3 ANÁLISIS DE SENSIBILIDAD

Se consideraron dos sensibilidades diferentes (relativas para poder comparar), que dependen de la secuencia de variables manipuladas aplicada $\mathbf{u}(t)$ (es decir, del experimento) y del punto θ considerado en el espacio paramétrico:

a) Las de la función objetivo a minimizar respecto de los parámetros j . Por diferencias finitas según (11) será la forma de cálculo habitual:

$$\frac{\partial J}{\partial \theta_j} \approx \theta_{j, \text{inicial}} \cdot \frac{J(\theta_j + \Delta \theta_j) - J(\theta_j)}{\Delta \theta_j} \quad (11)$$

aunque también pueden obtenerse analíticamente por integración numérica según (12):

$$\frac{\partial J}{\partial \theta_j} = 2 \sum_i \sum_{t=1}^N \gamma_i [\hat{y}_i(\mathbf{u}(t), \boldsymbol{\theta}, t) - y_i(t)] \frac{\partial \hat{y}_i(\mathbf{u}(t), \boldsymbol{\theta}, t)}{\partial \theta_j} \quad (12)$$

donde las parciales de las respuestas del modelo respecto de cada parámetro $\partial \hat{y}_i / \partial \theta_j$ pueden obtenerse derivando respecto a $\boldsymbol{\theta}$ las ecuaciones del modelo (6) y teniendo en cuenta que $\partial \mathbf{u} / \partial \boldsymbol{\theta} = 0$ resulta (13):

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right) &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \hat{\mathbf{y}}}{\partial \boldsymbol{\theta}} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}} \end{aligned} \quad (13)$$

siendo las incógnitas $\partial \mathbf{x} / \partial \boldsymbol{\theta}$. Integrando este sistema (13) de ecuaciones diferenciales junto a las ecuaciones del modelo (6), es posible obtener la evolución en el tiempo de $\partial \hat{\mathbf{y}} / \partial \boldsymbol{\theta}$ y por tanto las sensibilidades $\partial J / \partial \theta_j$ según (12).

b) Las sensibilidades de las respuestas del modelo (salidas) respecto de los parámetros según (14):

$$s_{ij} \approx \frac{\theta_{j, \text{defecto}}}{\bar{y}_i} \cdot \frac{\hat{y}_i(t, \theta_j + \Delta \theta_j) - \hat{y}_i(t)}{\Delta \theta_j} \quad (14)$$

Para obtener una medida cuantitativa de la importancia de cada parámetro individual j sobre cada salida i se utiliza el siguiente valor δ_{ij}^{msqr} – media cuadrática– integrado en el tiempo (15):

$$\delta_{i,j}^{msqr} = \sqrt{\frac{1}{N} \sum_{k=1}^N s_{ij}^2(k)} \quad (15)$$

En las Figuras 4 y 5 se muestran respectivamente las dos salidas del sistema del depósito para cambios en uno solo de los parámetros según el experimento de las Figuras 2 y 3, lo que permite hacerse una primera idea gráfica de la sensibilidad.

En la Tabla 1 se muestran como ejemplo los valores numéricos de las sensibilidades de las dos salidas y la función objetivo J para los 4 parámetros desconocidos del depósito.

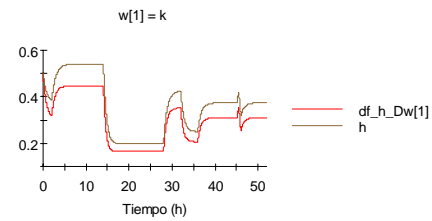


Figura 4: Sensibilidad de salida 1 vs. θ_1 .

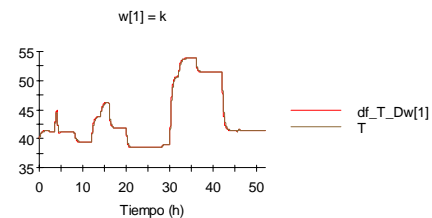


Figura 5: Sensibilidad de salida 2 vs. θ_1 .

Tabla 1: Sensibilidades respecto de los 4 parámetros a estimar en el modelo del depósito.

Parámetro	$\delta_{1,j}$	$\delta_{2,j}$	$\partial J / \partial \theta_j$
θ_1	1.54	0.04	344
θ_2	0	0.04	-7.3
θ_3	0.06	0.02	-1.1
θ_4	0	0.17	-34

En el análisis efectuado [6] se comprobó cómo ambas sensibilidades conducen en general a iguales conclusiones, aunque con ligeros matices ligados a las definiciones respectivas de las mismas. Las de las salidas $\delta_{i,j}^{msqr}$ apenas varían de θ_{inicial} a $\boldsymbol{\theta}^*$, pero no tienen en cuenta los pesos γ_i que ponderan las distintas salidas y que sí intervienen en la función objetivo a minimizar. Las de la función objetivo respecto de cada parámetro $\partial J / \partial \theta_j$ sí consideran los pesos que ponderan las distintas salidas i , pero presentan el inconveniente de depender demasiado de la posición relativa entre los datos experimentales y el modelo simulado para un $\boldsymbol{\theta}$ dado, por lo que pueden inducir a confusión cuando una salida se ajusta mucho mejor que otra u otras a los datos experimentales para un $\boldsymbol{\theta}$ en particular.

2.4 ANÁLISIS DE IDENTIFICABILIDAD

El orden de importancia de los parámetros obtenido a partir del análisis de las sensibilidades da una idea del efecto de cada parámetro –considerado individualmente– sobre las salidas. Sin embargo, es necesario estudiar también la influencia conjunta de los parámetros, pues puede ocurrir que el efecto sobre la salida provocado por un cambio en un parámetro se anule por un cambio en otro parámetro hecho simultáneamente. Se dice entonces que hay un

cierto grado de colinealidad en las sensibilidades de las salidas frente a los parámetros, lo que dificulta la identificación del conjunto de parámetros.

La identificabilidad es una propiedad estructural que depende de cómo los parámetros aparecen en el modelo, pero también de las medidas disponibles. Este aspecto puede mejorarse con un buen diseño de experimentos, pero a veces no puede hacerse nada. En cualquier caso, es bien sabido que la colinealidad no afecta negativamente a las predicciones del modelo, pero sí influirá en la precisión de la estimación –el valor más o menos exacto de los estimadores obtenidos–.

Se analizaron dos métodos para evaluar la identificabilidad de conjuntos de parámetros:

a) El método de Brun et al. (2002). Se basa en la dependencia lineal de subconjuntos de columnas de la matriz de sensibilidades S construida a partir de δ_{ij}^{msqr} . Brun define el índice de colinealidad (16):

$$\varphi_K = \frac{1}{\min_{\|\beta\|=1} \|\tilde{S}_K \beta\|} = \frac{1}{\sqrt{\lambda_K}} \quad (16)$$

donde \tilde{S}_K es una submatriz $n \times k$ de \tilde{S} que contiene las columnas correspondientes a los parámetros del subconjunto K , siendo n el número de salidas medidas y k el número de parámetros del subconjunto K ; \tilde{S} es la matriz de sensibilidades reescalada o normalizada, cuyas columnas se calculan según (17) y donde λ_K es el menor valor propio de $\tilde{S}_K^T \tilde{S}_K$.

$$\tilde{\delta}_j = \frac{\delta_j}{\|\delta_j\|} \quad (17)$$

b) El método de la matriz de información de Fisher (FIM). Como medida de la identificabilidad de los parámetros de un modelo con un experimento dado utiliza un índice de singularidad de matrices como el índice de condicionamiento, que se calcula como el cociente entre el máximo y el mínimo valor propio de la matriz FIM.

Valores altos de los índices de Brun o Fisher indican que las matrices son casi singulares, y por consiguiente el conjunto de parámetros θ no presentará una buena identificabilidad. El valor mínimo es 1 en ambos casos. En la Tabla 2 se muestran los resultados para algunos subconjuntos significativos del modelo del depósito.

Se comprobó [6] cómo ambos métodos, a pesar de proporcionar información análoga, pueden

complementarse en algunos casos particulares. El método de Brun puede conducir a colinealidades anormalmente elevadas de dos parámetros –a pesar de que éstos sean perfectamente identificables de acuerdo con el modelo– si los dos parámetros afectan de forma similar a las salidas (θ_2 θ_4). El método de Fisher por el contrario presenta la ventaja de permitir evaluar mejor la identificabilidad de acuerdo con el modelo y el experimento considerado; pero a su vez puede dar valores anormalmente altos de colinealidad cuando las sensibilidades relativas de los parámetros sean muy distintas en magnitud (θ_1 θ_2), al estar entonces la matriz FIM mal condicionada. De ahí que ambos métodos puedan ser utilizados de forma complementaria.

Tabla 2: Identificabilidad, modelo del depósito.

Subconjuntos parám.	Brun	Fisher
θ_1 θ_2 θ_3 θ_4	∞	1240
θ_1 θ_2	1.01	384
θ_1 θ_3	5.4	434
θ_1 θ_4	1.01	21
θ_2 θ_4	32000	59

Por último, se analizó [6] cómo las desviaciones típicas adimensionales σ_j calculadas a partir de la FIM pueden también utilizarse alternativamente para hallar una clasificación de los parámetros y seleccionar uno o varios subconjuntos a estimar, ya que los errores relativos de estimación (desviaciones típicas adimensionales usadas en el cálculo de regiones de confianza) que predice la FIM de forma aproximada para los parámetros dependen tanto de las sensibilidades individuales como de la identificabilidad dentro del conjunto.

2.5 ESTIMACIÓN POR OPTIMIZACIÓN

Se estimarán aquellos parámetros que presenten una sensibilidad elevada o considerable, nunca aquellos que presenten sensibilidades despreciables. Y en cuanto a la identificabilidad, interesa que el subconjunto de parámetros a estimar presente una baja colinealidad. Si la colinealidad es alta, habrá muchas combinaciones de valores θ^* que conducirán al mismo valor de J aproximadamente. No obstante, puede decidirse estimar un subconjunto de parámetros con elevada colinealidad, ya que normalmente estimar un número mayor de parámetros hace que el ajuste a los datos experimentales sea mejor. Aunque si el número de parámetros es considerable, lo lógico será seleccionar un conjunto con una buena identificabilidad para reducir el número de parámetros a estimar y facilitar así la tarea al algoritmo de optimización.

Tras seleccionar uno o varios subconjuntos de parámetros, la estimación se efectúa por optimización dinámica de acuerdo con una función multiobjetivo correspondiente a un criterio de mínimos cuadrados ponderados. Los datos experimentales no se filtraron dado que el ruido no afecta negativamente a la optimización y en modelos no lineales no está comprobado que conduzca a mejores estimas.

Tabla 3: Parámetros estimados e intervalos de confianza, cuando se estimaron 4 y 2 param.

Parám.	$\theta_{inicial}$	$\hat{\theta}_{4 param}$	$\hat{\theta}_{2 param}$
θ_1	0.72	(0.6195 \pm 0.0004)	(0.6195 \pm 0.0005)
θ_2	34	(52.4 \pm 1.2)	
θ_3	0.47	(0.465 \pm 0.007)	
θ_4	7.0	(9.69 \pm 0.11)	(11.08 \pm 0.09)

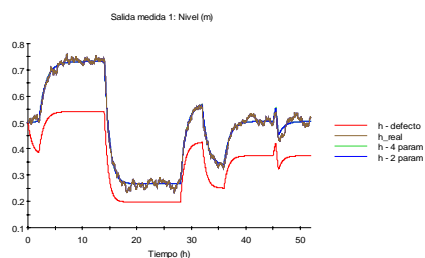


Figura 6: Salida medida 1.

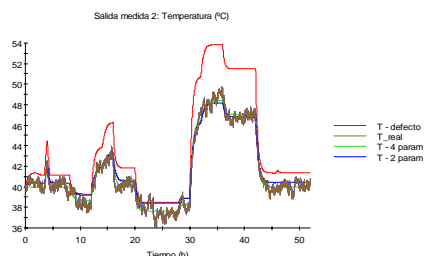


Figura 7: Salida medida 2.

En la Tabla 3 se muestran los estimadores (no escalados y con dimensiones) obtenidos cuando se estimaron 4 y 2 parámetros en el caso del depósito. Las Figuras 6 y 7 muestran la diferencia entre el ajuste del modelo a los datos experimentales antes y después de la estimación, cuando se seleccionaron subconjuntos de 4 y 2 (θ_1 θ_4) parámetros a estimar.

3 RESOLVEDORES EN EL ENTORNO ECOSIMPRO®

El entorno de modelado y simulación empleado EcosimPro® presenta muchas cualidades que le confieren una gran versatilidad y potencia.

Para la estimación de parámetros se analizó el funcionamiento de una herramienta general de optimización no lineal (NLP) a utilizar como resolutor en problemas de optimización planteados en EcosimPro®, una rutina NAG® de minimización NLP que implementa un método de programación cuadrática secuencial SQP.

Dicho algoritmo utiliza un método determinista. Presenta la ventaja de la rapidez frente a un método heurístico, pero en general no garantiza que el óptimo hallado sea el global. Otras consideraciones generales a tener en cuenta cuando se trabaja en optimización son el necesario escalado de las variables de decisión, así como la importancia de inicializar y delimitar convenientemente las variables de decisión.

3.1 RUTINA DE LA LIBRERÍA NAG®

Para enlazar la rutina de optimización NAG® con EcosimPro® se empleó una llamada a la misma desde el experimento de EcosimPro®. Y se analizó la conveniencia de proporcionar a la rutina los gradientes analíticos de la función objetivo a optimizar respecto de las variables de decisión ($\partial J / \partial \theta$), gradientes obtenidos por integración numérica en EcosimPro® según (12).

El interés en analizar la conveniencia de proporcionar las derivadas analíticas integradas numéricamente – frente a la opción habitual de que sea el optimizador quien las estime como cociente de incrementos a partir de evaluaciones de la función objetivo– es debido a que uno de los puntos más críticos para la búsqueda del óptimo hace referencia al cálculo de las derivadas. Y es habitual en este tipo de problemas de optimización donde la función objetivo a minimizar se calcula tras una simulación que alrededor del 90% del tiempo de computación sea dedicado a la simulación del modelo (bien para obtener el valor de la función objetivo o sus gradientes), especialmente cuando el modelo tiene cierto grado de complejidad, por lo que es importante tratar de aumentar la eficiencia en la búsqueda de la solución.

Se evaluaron los resultados para distintas condiciones iniciales y límites en los parámetros, de acuerdo con la robustez (valor final alcanzado J^*), el número de iteraciones del algoritmo y el tiempo total de computación. Por otra parte, es sabido que la precisión del simulador ha de ser mayor que la precisión del resolutor para que el algoritmo de optimización proporcione unos resultados fiables cuando la obtención de la función objetivo J involucra la resolución de sistemas de ecuaciones y simulación; un valor de dos órdenes de magnitud de diferencia suele ser suficiente. Por lo que igualmente se realizaron diferentes pruebas modificando

conjuntamente las precisiones de cálculo de la rutina de optimización NAG[®] y de EcosimPro[®].

La conclusión obtenida es que el óptimo encontrado cuando se proporcionan los gradientes analíticos a la rutina NAG[®] nunca supera ni en el valor hallado ni en el tiempo empleado al caso por defecto en que es la propia rutina quien evalúa esos gradientes por diferencias finitas. Puede por tanto extrapolarse a otros problemas de similares características y concluirse que no es conveniente proporcionar al optimizador los gradientes analíticos obtenidos por integración numérica en EcosimPro[®]; son costosos de calcular en tiempo y recursos y no conducen a una exactitud significativamente mayor en el óptimo alcanzado. Así pues, será más eficaz que la propia rutina NAG[®] de optimización los estime por diferencias finitas mediante evaluaciones de la función objetivo, perturbando las variables de decisión en la medida adecuada para que las derivadas sean suficientemente precisas.

4 VALIDACIÓN DE MODELOS

Tras la estimación de parámetros, la etapa posterior en el modelado de procesos es la validación del modelo resultante.

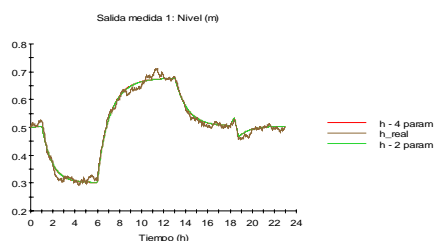


Figura 8: Validación, salida 1.

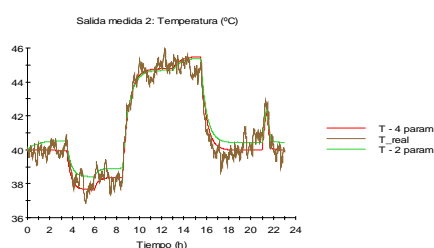


Figura 9: Validación, salida 2.

La validación es crítica dado que permite evaluar la credibilidad del modelo de acuerdo con el propósito determinado para el que se construye. La confianza en un modelo se obtiene a partir de resultados positivos en un conjunto de test. Si los resultados no fueran satisfactorios podría ser necesario repetir bien la estimación de parámetros —a partir de otros datos experimentales— o bien todo el proceso de establecimiento de hipótesis y formulación del

modelo. Se utilizaron distintas técnicas de validación del modelo propuesto, de acuerdo con un procedimiento sistemático [8], de las cuales, por problemas de espacio, sólo mencionamos la comparación con datos experimentales.

Para un experimento diferente al de parametrización, el modelo obtenido conduce a los resultados que se muestran en las Figuras 8 y 9. Se observa cómo el modelo estimando 4 parámetros permite un mejor ajuste que estimando 2 parámetros en la salida 2, pero la diferencia puede no ser significativa.

Agradecimientos

A Fernando Carbonero, de Empresarios Agrupados S.A., quien proporcionó valiosas referencias.

A ISA España, por todas las inestimables oportunidades que ofrece a los miembros de las Secciones de Estudiantes adscritas.

Referencias

- [1] Brun, R., et al., (2002) “Practical identifiability of ASM2d parameters- systematic selection and tuning of parameter subsets”, *Water Research* 36, pp. 4113-4127.
- [2] Chen, H., Kremling, Allgöwer, F., (1995) “Nonlinear predictive control of a benchmark CSTR”, *Proceedings of 3rd European Control Conference*, Roma.
- [3] Dueñas Díez, M., (2004) “Population balance modeling and passivity-based control of particulate processes, applied to the Silgrain[®] process”, Tesis Doctoral, Telemark University College.
- [4] Edgar, T., Himmelblau, D., Lasdon, L., (2001) “Optimization of chemical processes”, McGraw-Hill, Boston.
- [5] Empresarios Agrupados International, (2004) “EcosimPro Guides”, Madrid.
- [6] Gómez, E., (2006) “Optimización dinámica en EcosimPro[®]. Aplicación a la estimación de parámetros y optimización de un reactor”, Proyecto fin de carrera, Univ. de Valladolid.
- [7] Ljung, L., (1987) “System identification: theory for the user”, Prentice-Hall, New Jersey.
- [8] de Prada Moraga, C., (2004) “Validación de modelos”, Jornadas de Modelado y Simulación de procesos continuos organizadas por CEA-IFAC, Valladolid.

jal.06

XXVII jornadas de automática almería

INSTITUCIONES ORGANIZADORAS



EMPRESAS PATROCINADORAS



COLABORADORES Y PATROCINADORES DE PREMIOS DE GRUPOS TEMÁTICOS

